State-Based Decision Support for Condition Shifting of Multimodal Continuous Processes

by

Francois Frieder Noelle

Thesis presented in partial fulfilment

of the requirements for the Degree

of

MASTER OF ENGINEERING

(EXTRACTIVE METALLURGICAL ENGINEERING)

in the Faculty of Engineering

at Stellenbosch University

Supervisors

Dr. T.M. Louw

Prof. S.M. Bradshaw

Prof. L. Auret

March 2021

DECLARATION

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: *November 2020*

PLAGIARISM DECLARATION

1. Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.

2. I agree that plagiarism is a punishable offence because it constitutes theft.

3. I also understand that direct translations are plagiarism.

4. Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism.

5. I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Initials and surname: F.F. Noelle

Date: March 2021

ABSTRACT

Automated distributed control systems are able to keep controlled variables at their set points and thus are able to maintain process operation within an operating mode (quasi-steady state). These automated systems may however not be informed of complex issues within the process, as a result, human supervision is required within the control loop. These supervisors are therefore required to perform manual actions to allow a process to settle to safer or more profitable operating conditions. Modern industrial continuous processes thus undergo frequent state shifts either due to set point changes or sustained disturbances. Fundamental process models may assist supervisors in the evaluation of process conditions beyond their empirical experience, however, the development of such models is difficult and may require significant effort. Due to the existence of distributed control systems, many process plants have large historical databases of past sensor measurements. Data-driven approaches to process monitoring are therefore applicable.

This investigation aims to discover the various steady states conditions (process modes), their economic performance, and switching conditions from the continuous process's multimodal historical data. This knowledge can then be leveraged to provide decision support to supervisors, which would allow them to operate the process more profitably. Since historical process data is mostly not classified into its various states, an unsupervised data-driven approach is imperative. Specifically, the approach is developed and evaluated on synthetic multimodal data obtained from a propylene glycol reactor simulation, and finally implemented on actual industrial milling circuit data. The developed state-based decision support model made use of principal component analysis, stationarity analysis, K-means clustering, Gaussian mixture models, and key performance indicators in an integrated manner.

Stationarity analysis was able to effectively detect and thus remove transient states from the simulated CSTR data, however, considerable process knowledge was required in setting the algorithm hyperparameters. K-means clustering was utilized to provide initial parameter estimates for the Gaussian mixture model fitting. The best model configurations were selected based on the lowest Bayesian information criterion, however, the suggested best model usually overfit the data. Additional model refinement was therefore required such that each process mode or steady-state was described by a single Gaussian within the model. These refining procedures made use of the data sequence, Euclidean distance between Gaussians, and their prior probabilities. The results showed that even if transient states are not removed prior to the analysis, relatively good monitoring performance can be achieved with the developed approach. Further, contribution plots were utilised to identify the key variables that may have resulted in a transition. As a result, useful decision support can be provided to process supervisors.

An algorithm was developed which summarises high dimensional correlated historical process data into a two-dimensional process state "map", which can effectively assist supervisors in navigating complex multimodal continuous processes. Further, expert knowledge can continually be leveraged to refine the process map and its corresponding model since the data-driven approach emphasizes human-machine interactions. The decision support system worked well on simulated CSTR data, however, more advanced procedures are required to "diagnose" the causes of transitions within industrial process data.

OPSOMMING

Geoutomatiseerde verspreide beheerstelsels kan veranderlikes by hul setpunte beheer en kan prosesbedryf binne ’n bedryfsmodus handhaaf (quasi-bestendige toestand). Hierdie geoutomatiseerde stelsels kan egter nie kennis neem van komplekse probleme binne die proses nie, en as ’n gevolg, word menslike toesig benodig om aksies met die hand uit te voer om ’n proses na veiliger of meer winsgewende bedryfskondisies te bring. Moderne industriële aaneenlopende prosesse gaan dus gereelde toestand veranderinge deur as gevolg van setpuntveranderinge, of aanhoudende steuringe. Fundamentele prosesmodelle kan toesighouers assisteer in die evaluasie van proseskondisies verder as hul empiriese ondervinding, maar die ontwikkeling van sulke modelle is moeilik en mag beduidende moeite vereis. As gevolg van die bestaan van verspreide beheerstelsels, het baie prosesaanlegte groot historiese databasisse van vorige sensormates. Datagedrewe benaderinge tot prosesmonitering is daarom toepaslik.

Hierdie ondersoek mik om die verskillende bestendige toestande (prosesmodus), hul ekonomiese doeltreffendheid, en omruilkondisies van die aaneenlopende proses se multimodale historiese data, te ontdek. Hierdie kennis kan dan gebruik word om besluitondersteuning aan toesighouers te verskaf, wat hulle dan sal toelaat om die proses meer winsgewend te bedryf. Aangesien historiese prosesdata meestal nie in sy verskeie toestande geklassifiseer is nie, is ’n ongekontroleerde datagedrewe benadering noodsaaklik. Meer spesifiek, die benadering is ontwikkel en geëvalueer op sintetiese multimodale data verkry van ’n propileen-glikolreaktorsimulasie, en uiteindelik geïmplementeer op ’n werklike industriële malery se stroomdata. Die ontwikkelde toestand-gebaseerde besluit ondersteuning model het gebruik gemaak van hoofkomponentanalise, stasionariteitanalise, K-gemiddelde groepering, Gauss-mengselmodelle, en sleutel doeltreffendheidsindikators op ’n geïntegreerde manier.

Stasionariteitanalise het oorgangstoestande effektief opgespoor en dus van die gesimuleerde KGTR-data verwyder, maar aansienlike proseskennis was vereis om die algoritme hiperparameters te stel. K-gemiddelde groepering is gebruik om aanvanklike parameterberamings te verskaf vir die Gauss-mengselmodelpassing. Die beste modelkonfigurasies is gekies gebaseer op die laagste Bayesiaanse informasie kriteria, maar die voorgestelde beste model het gewoonlik die data oorgepas. Addisionele modelverfyning was daarom nodig sodat elke prosesmodus of bestendige toestand deur ’n enkel Gauss-kromme binne die model beskryf kon word. Hierdie verfyningprosedures het gebruik gemaak van die datareekse, Euklidiese afstand tussen Gauss-krommes, en hul vorige waarskynlikhede. Die resultate het gewys dat selfs as oorgangstoestande nie voor die analise verwyder is nie, kan relatief goeie moniteringsdoeltreffendheid bereik word met die ontwikkelde benadering. Verder, verspreidingsplotte is gebruik om die sleutel veranderlikes te identifiseer wat ’n oorgang tot gevolg kon hê. As resultaat, kan bruikbare besluitondersteuning verskaf word aan prosestoesighouers.

’n Algoritme is ontwikkel wat hoë dimensionele, gekorreleerde, historiese data opgesom het en in ’n twee-dimensionele prosestoestand gekarteer het, wat effektiewelik gebruik kan word om toesighouers te assisteer om komplekse multimodale aaneenlopende prosesse te navigeer. Verder, deskundige kennis kan deurentyd gebruik word om die proseskaart en sy korresponderende model te verfyn aangesien die

datagedrewe benadering mens-masjien interaksie beklemtoon. Die besluit ondersteuning sisteem het goed gewerk op gesimuleerde KGTR-data, maar meer gevorderde prosedures is nodig om die oorsake van oorgange binne industriële prosesdata te diagnoseer.

ACKNOWLEDGEMENTS

I would like to thank the various people and institutions that made this research possible:

- **Dr. Tobi Louw**, **Prof. Steven Bradshaw** and **Prof. Lidia Auret** for allowing me to investigate their unique and useful concept, which has definitely changed the course of my career. Further, for encouraging me, providing me with insight, advice, and finally for continually correcting my course.
- **Dr. De Villiers Groenewald** and **Anglo American Platinum** for providing me with the opportunity, support, industrial data and advice.
- **Dr. Alexey Cherkaev** and the rest of the **Machine Learning Research Group** for the good times and exchange of knowledge.
- **My Family** for giving me all the support, love, and trust I could need.
- **My friends**, for listening to me ramble about converging to local maximums, transient states and imbalanced data.

TABLE OF CONTENTS

NOMENCLATURE

Symbols	Description
α	Significance
D	Distance
$\mathbf{E}$	Residual Matrix
ϵ	Random Noise
$\mathbf{EW}$	Switching Frequency Matrix
$\emptyset$	Variable Loadings
F	Fraction within interval
$g(\cdot)$	Multivariate Gaussian Distribution
$L(\cdot)$	Likelihood
λ	Eigenvalue
$\mathbf{\Lambda}$	Eigenvalue Matrix
m	Drift Component
$\mathbf{MAP}$	Shift Condition Matrix
μ	Univariate Mean
$\boldsymbol{\mu}$	Multivariate Mean Vector
n	Samples within Dataset
n	Window Size
σ	Standard Deviation
μ	Mean
n	Window Size
$p(\cdot)$	Probability Density
$P(\cdot)$	Posterior Probability
P	Midpoint Performance
$\mathbf{P}$	Principle Component Loading Matrix
ϕ	Autoregressive Coefficient
Q	Residual Threshold
r	Sample Residual for Variable
$\boldsymbol{r}$	Sample Residual Vector
$\mathbf{S}$	Estimated Covariance
σ	Standard Deviation
$\mathbf{\Sigma}$	Covariance Matrix

t_{crit}	Critical Test Statistic (Student's t-test)
T^2	Hotelling's T Squared
θ	Steady State Detection Threshold
$\boldsymbol{\theta}$	Gaussian Parameters Vector
$\boldsymbol{\Theta}$	Gaussian Mixture Model Parameters
$\mathbf{v}$	Eigenvector
$\mathbf{V}$	Eigenvector Matrix
w	Gaussian Weight
x	Univariate Sample Sample
$\hat{x}$	Normalized Sample
$\mathbf{x}$	Multivariate Sample Vector
$\mathbf{X}$	Training/Testing Dataset
y	Univariate Binary Stationarity
Y	Multivariate Extension Binary Stationarity
z	Principle component Score
$\mathbf{z}$	Sample Principal Component Score Vector
$\mathbf{Z}$	Principal Component Score Matrix
Superscripts and Subscripts	**Description**
α	Threshold
Adj	Detected Transients Removed
c	Specific Component (Cluster)
CVE	Cumulative Variance Explained
e	Common Cause
i, j, z	Identifying Indices
k	Reduced Dimension
K	Number of Gaussians or Clusters
m	Number of Variables in Dataset
M	Maximum
n	Number of Samples
s	Iteration Repetition
s	Stationary Period Duration
t	Sample from Time Series
VE	Variance Explained

Abbreviations	Description
BIC	Bayesian Information Criterion
CSTR	Continuous Stirred Tank Reactor
DPCA	Dynamic Principle Component Analysis
EM	Expectation Maximisation
FAR	False Alarm Rate
FN	False Negative
FP	False Positive
FTR	False Transient Rate
GMM	Gaussian Mixture Models
GPR	Gaussian Process Regression
HMM	Hidden Markov Model
KDE	Kernel Density Estimate
KPI	Key Performance Indicator
LDA	Linear Discriminant Analysis
LPP	Locality Preserving Projections
MSPC	Multivariate Statistical Process Control
MsPCA	Multiple Set PCA
MTR	Missed Transient Rate
NLLP	Negative Log-Likelihood Probability
PC	Principal Component
PCA	Principle Component Analysis
SOM	Self Organising Maps
SOP	Standard Operating Procedure
SPE	Squared Prediction Error
SSD	Steady State Detection
TAR	True Alarm Rate
TP	True Positive

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Background

The main objective of process control is to maintain a process at the desired operating condition efficiently and safely, while satisfying environmental and product quality requirements (Seborg, Edgar and Mellichamp, 2004). Automatic process control may however fail in eliminating the course of poor operation, especially when these situations are not accounted for within process models (Wang, 1999). During these situations, human supervisors are required to make key decisions and adjustments that would allow a process to reattain optimal conditions. Supervisors must therefore develop an understanding of the process performance in the long and short term while supervising the control strategy. This understanding can be used to identify problems in current operation, deteriorating equipment, and operating regions of improved efficiency (Wang, 1999). This implies that human supervisors are an integral part of the overall control system and should therefore be provided with the means to carry out their role effectively (Wang, 1999).

Human supervisors, however, may have trouble identifying issues within high dimensional data, therefore assistance should be provided to assimilate data (Wang, 1999). Statistical process control methods such as Shewhart charts have been used to monitor key product variables to determine the occurrence of an event and its cause. Most of these methods are however only able to analyse a small number of variables and are no longer acceptable for modern processes (Wang, 1999). Variables such as pressure, temperature, composition, flow, and vessel levels are sampled often, producing massive amounts of data that need to be analysed. The large amount of data produced at process plants is multivariate and correlated, therefore needs to be analysed simultaneously. The need to provide computer assistance in assimilating data has become a major concern and therefore automatic data analysis systems form a vital part of integrated control systems (Wang, 1999).

1.2 Problem Statement

Contributing to the complexity of a supervisor's role is the fact that most continuous chemical process plants operate at a multitude of different steady states. Various reasons exist why a plant would operate at different modes, processes may switch modes due to changes in the feed composition, product slate or maintenance operations (Srinivasan, Viswanathan and Vedam, 2005). Such mode switches are termed transitions, which require a considerable amount of operator or supervisor involvement to be achieved effectively (Srinivasan, Viswanathan and Vedam, 2005). Operators are required to follow predefined standard operating procedures (SOP) that allow the plant to settle to different steady states, which may include reconfiguring controllers with different settings or starting/stopping process units (Srinivasan, Viswanathan and Vedam, 2005). During transitions the probability of abnormal events and off specification products arises (Nimmo, 1993), therefore it is imperative to manage these transitions efficiently (Srinivasan, Viswanathan and Vedam, 2005). Transitions may however also occur due to actions not attributable to supervisor actions, such as disturbance-driven transitions.

A process state can be described by either a mode or a transition. Modes are defined as an operation where all variables remain in a quasi-steady state in which the constituent variables only vary within a narrow range due to noise, instrumentation faults, and controller action (Srinivasan, Viswanathan and Vedam, 2005). A mode's key variables therefore only vary within a narrow range. Each mode has different characteristics, some may be more safe or optimal than another. Another fact that needs to be taken into consideration is that mode switching constraints exist. Due to operating and equipment constraints, not every mode is reachable through another (Afzal, Tan and Chen, 2017).

A transition corresponds to discontinuities in plant operation, such as a change in set points or the occurrence of a sustained disturbance (Srinivasan, Viswanathan and Vedam, 2005). At least one of the mode's process variables undergoes considerable change, however not all constituent variables vary during transitions. Usually, transients occur due to an operator-induced action. Human supervisory staff have to take various complex control decisions to induce a transition, during which less yield and poor product quality can be expected until the process attains the new steady state condition (Srinivasan, Viswanathan and Vedam, 2005). For example, in the refining industry product and feedstock changes occur frequently, usually operating on 10-20 different crudes on a regular basis, therefore resulting in mode changes occurring 3-5 times a week. These transitions last for approximately 4-8 hours and therefore could have consequent economic impact if the transitions are not performed effectively (Srinivasan, Viswanathan and Vedam, 2005). Each grade corresponds to separate mode; however, if the same grade is produced at a different throughput, variables such as flowrate change and therefore also corresponds to a mode change (Srinivasan, Viswanathan and Vedam, 2005).

With the extensive implementation of distributed control systems within process plants a window of new opportunity is available. Data availability and human-machine technologies are allowing for more effective optimization, monitoring and decision making (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019). Recently, extensive research has been directed towards monitoring processes, such that faults in operation can be identified and diagnosed rapidly. Little research has ,however, been conducted into the extension of the analysis of multimodal processes beyond monitoring (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019). The reformulation of production patterns (associated with safety and optimization) to account for the multimodality of large scale processes is a challenge that is still faced (Jiang, Li and Yin, 2018).

1.3 Aims and objectives

The aim of this investigation is to develop a data-driven decision support system that is able to determine the state a continuous multimodal process is in and how to get to different process modes. The system should be able to identify operating conditions that would maximize economic benefit, as well as the current relative performance. From historical data, the system should be able to identify procedures required to transition the process to a state of maximum economic benefit while taking into account mode shifting constraints. This information can then be summarized in the form of actionable advisories, supporting supervisory decisions.

Figure 1 displays an example sketch, as to how these actionable advisories could be provided. For example, here the process is currently in mode 1, which has an undesirable economic performance. Mode 4 is identified to have the best economic performance, thus it is suggested to shift to mode 4. Since mode 3 is only reachable from mode 1 via a disturbance-caused transition, the suggested procedure (SOP) would be to shift the process to mode 2, mode 5, and then mode 4 is finally attainable.

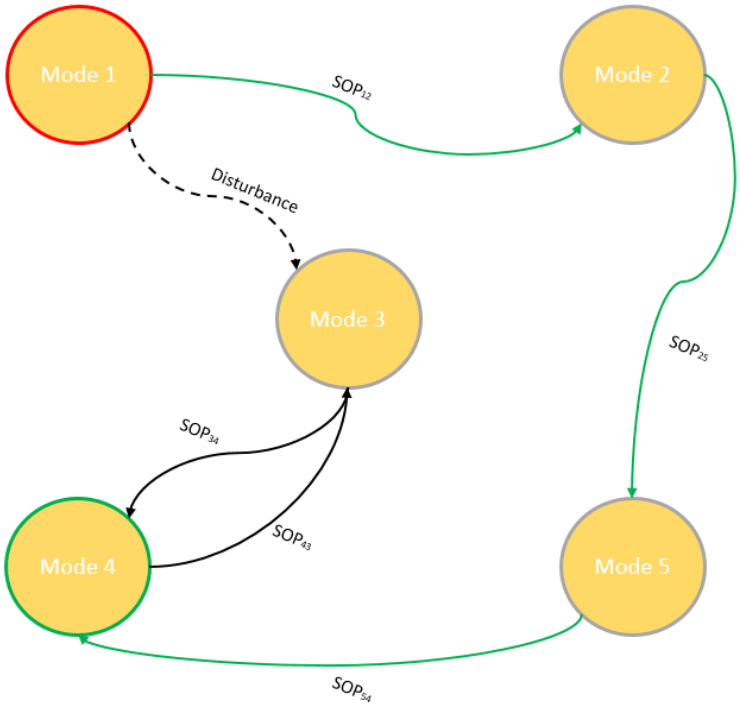


Figure 1: Sketch describing the form in which actionable advisories can be provided to supervisors

Thus, the analysis of multimodal processes is extended beyond monitoring, rather assisting human supervisors in navigating multimodal continuous processes such that more optimal conditions can be achieved. Such a system would enhance the modern control strategy which has been designed to take maximum advantage of automation yet still benefit from the intelligence of the human agent (Salvendy, 2012).

To achieve this broad aim, certain objectives have to be set.

1. Develop a state-based decision support approach that is able to discover and identify modes within data.
   - i) The approach should be able to discover modes within historical process data.
   - ii) From historical data, the approach should determine the procedures required to transition from one mode to another, if these procedures exist.
   - iii) On real-time data (testing data), the approach should be able to identify the current mode of operation and its economic optimality.
   - iv) The approach should suggest a more economically desirable mode (if it exists) while taking into account mode shifting constraints.

v) The approach should advise the procedures required to attain more desirable operating conditions.

vi) The approach should be able to provide these actionable advisories (iv and v) such that they are easily interpretable by human supervisors.

2. Evaluate the performance of the state monitoring approach on multimodal simulation data.
3. Determine the applicability of the state-based approach on historical industrial process data.

1.4 Approach

To achieve the stated objectives an unsupervised data-driven approach is utilized. Specifically, techniques of steady state analysis, dimensionality reduction, and clustering are implemented with the guidance of expert knowledge to produce a process "map". This map effectively summarizes all discovered knowledge. The developed state-based decision support system is evaluated on a simulated continuous stirred tank reactor (CSTR) producing Propylene Glycol at different process states. The developed techniques are finally implemented on an industrial process dataset to determine their applicability to actual process data.

2 LITERATURE REVIEW

2.1 Process Plant Control

An understanding as to how processes are controlled is required prior to the discussion of data analysis techniques. This understanding allows for the identification of information that may be relevant to the supervisory staff, thus providing advisories that could improve in process operation. Further, this understanding can also assist in the model development itself, guiding key model design decisions.

2.1.1 Control Objectives

Control structures are required to maintain or improve plant operation by maintaining controlled variables at their desired values when disturbances occur as well as responding to changes in the desired values (Marlin, 1995). To allow the plant to operate smoothly, various key control objectives have been formulated. These objectives entail process safety, environmental protection, equipment protection and product quality. Three additional objectives must be achieved simultaneously otherwise plant operation will be unprofitable or dangerous (Marlin, 1995).

1. Smooth operation and production rate: Processes are operated with tight regulation on controlled variables and smooth adjustments of manipulated variables. It is thus a key objective for process plants to be operated such that they can be classified as modes.
2. Profit: The goal of most process plants is to achieve a profit. Sometimes additional degrees of freedom exist after satisfying the previous control objectives (safety etc.) to maximize profits.
3. Monitoring and Diagnosis: This objective is usually performed by two groups. The operators ensure the immediate safe operation of a plant, whereas the supervisors monitor the long term performance of a process. Both are required to intervene and restore plants to acceptable performance if changes occurred. It should therefore be realized that both these monitoring approaches require a human in the loop that would address complex issues that automated systems do not take into account. These people are therefore required to make important decisions that will be implemented manually, allowing the plant to operate safely and profitably in terms of long term performance.

Regulatory control objectives assist in keeping controlled variables within their bounds or specified set points. Economic objectives on the other hand require for the optimization of control actions (Morari and Stephanopoulos, 1980). Once the key variable's variation has been reduced to maintain operation within a mode, the desired value of the controlled variable can be adjusted to increase profit thus inducing a mode transition (Marlin, 1995).

2.1.2 Supervisory Control

Most automated supervisory control strategies require an understanding of the dynamics within complex processes, in the form of a model. These models are either formulated from fundamental principles, empirical data or a combination of both. It is crucial that the complex interactions between multiple control strategies is determined (Marlin, 1995).

2.1.2.1 Development of a Process Plant Model

The development of an effective automated process depends on two key aspects. Firstly improvements in the automation of a plant can be made during the design phase of the process. Designing equipment and control structures that ensure good plant controllability are therefore required. Secondly, knowledge of the dynamic behavior of the plant is important for the implementation of automation (Marlin, 1995).

Fundamental models allow for extrapolation beyond regions of immediate empirical experience, allowing staff to evaluate changes in conditions within the plant. The modelling procedure is mostly done in two steps. In the first step model development occurs, during which model goals are defined. Here, the required information is collected and the model is formulated. In the second step, the model is simulated, where model solutions are determined, after which the model results are analyzed and validated (Marlin, 1995).

The model procedure should be matched with the problem goals. The modelling goals should be based on the type of information required and its application (Marlin, 1995). Model goals have a great effect on the accuracy of the developed model. Model formulation is based on a set of assumptions, such that the modelling goals are satisfied. A balance between a complex model which may capture variable interactions more effectively and a simpler model from which a solution can be computed more easily (Marlin, 1995), should be kept in mind when making assumptions. Only through careful analysis of model solutions can it be reassured that a developed model reflects realistic situations (Marlin, 1995).

2.1.2.2 Model-Based Optimization

In order to realize a process's economic objectives a set of operating conditions have to be chosen that would optimize determined objective functions. These operating conditions are the characteristics that define a mode, however not all modes within historical data are optimal. Optimization models, therefore, require objective functions, decision variables, and constraints (Winston, 2003). The objective function is the function that should be minimized or maximized to satisfy the plant's economic objectives. The decision variables are the variables within the plant that are under the human supervisor's control, therefore the controlled variables. Constraints are the restrictions on certain decision/controlled variables (Winston, 2003), such as the mode switching constraints mentioned earlier. When setting control objectives a clear understanding of how the plant operating conditions are determined is required.

Firstly when optimization is performed, the region of possible operation needs to be defined. This is called the operating window, which is the range of possible steady-state values of process variables that can be achieved with the available equipment (Marlin, 1995). However, for multivariate systems, the determination of the operating window is complex since it is influenced by the interaction of the various process variables. A trial and error procedure involving many simulations of the integrated process model within the process constraints is required to capture the interactions between the various process variables from which the operating window can be determined (Marlin, 1995).

Even though any operation within the operating window is possible, these operating conditions may only satisfy the minimum plant goals. Great differences in plant profit exist depending on the operating conditions chosen within the window (Marlin, 1995). Current industrial practices indicate that optimal operating points switch from the intersection of one set of active constraints to another as process disturbances change with time (Morari and Stephanopoulos, 1980).

Optimizing control structures consequently should perform two important actions (Morari and Stephanopoulos, 1980). Firstly, they should identify and monitor the set of active constraints that determine the optimum operating point. And secondly, they should transition the process to this optimum by making process adjustments.

Model-based optimization should always account for inaccuracies within the model and its parameters. Three key aspects need to be considered when implementing optimizing control, therefore inducing a transition (Morari and Stephanopoulos, 1980).

1. A process should only be transitioned to an optimum mode when the disturbance is sustained or "slow" in relation to the faster process dynamics. Classification of the disturbances in terms of their frequency and economic impact is therefore essential when the decision to optimize is made. The decision to optimize is therefore guided by the classification of the disturbance according to its economic impact.
2. Transitioning to a new optimum mostly requires a sequence of set point changes. Determination of the sequence within the existing control structure is crucial for optimization.
3. All operating points should remain within the operating window. The constraints which define the optimum operating point (mode) or the active constraints should always be monitored.

2.1.2.3 Importance of Human Supervisory Control

Developed automated control strategies are able to maintain control objectives under normal conditions, but fail to maintain these objectives under abnormal conditions, especially when an abnormality is sudden and unexpected (McLeod, 2015). Abnormal events refer to states that are not accounted for within developed models.

It is during these situations where human supervisory staff is required to intervene the automated control system. As a result, human supervisory control has been redefined from a set of lower-level skill-based behaviors to a set of higher-level knowledge-based behaviors (Cummings, Bruni and Mitchell, 2010). Modern control strategies, therefore, implement computers for routine executions of control actions based on sensed feedback, on the other hand, implement humans for setting goals (Salvendy, 2012).

2.1.2.4 Reliance of Process Plants on Human Supervisory Staff

Supervisory staff generally have extensive experience with their processes, therefore are able to perform control "by feel" (Kuespert and McAvoy, 1994). Supervisory staff have to perform various complex tasks within a process, such as operation start-up. During start-up various operator actions have to be performed simultaneously to establish heat, mass and pressure balance within the different plant sections (Srinivasan, Viswanathan and Vedam, 2005). This experience is however highly internalized and is mostly held implicitly rather than in the form of explicit rules. Experts are often not able to convey their

techniques, and if explanations are offered they often contradict their actual actions due to difficulties expressing implicit conscious expertise (Kuespert and McAvoy, 1994).

This fact may pose certain challenges to the control objectives. An information gap between supervisory staff may lead to ineffective operations (Srinivasan, Viswanathan and Vedam, 2005). All staff may also not have the same level of skill (Kuespert and McAvoy, 1994), therefore inefficiencies may occur if certain human supervisors are not present. Due to the internalized fashion experience is kept, process plants are also highly dependent on specific individuals. Along with known "boom and bust" cycles that occur throughout various industries, the loss of a specific individual may result in large drops in process efficiency.

Fluctuations in control quality due to the various mentioned reasons therefore result in fluctuations in process conditions, creating sub-optimal situations (Kuespert and McAvoy, 1994). Therefore many abnormal situations within processes arise due to human error (Nimmo, 1995). If the expertise of skilled operators were captured, processes would realize a significant improvement in safety, quality, productivity, and correspondingly economic benefit (Kuespert and McAvoy, 1994). Knowledge which is specific to individuals can be used to improve their and other staff skills through training, as well as generic knowledge that can be used to improve process equipment and control systems. Standardized strategies for handling abnormal situations can be developed further improving future process operation (Sebzalli, Li and Chen, 2000). Such systems are known as expert systems, which make use of logic rules to carry out heuristic reasoning. However, as mentioned, acquiring this knowledge from complex processes may be difficult (Wang, Chen and Yang, 1997).

Modern industrial plants have employed data acquisition equipment to record process measurements, including the actions of supervisory staff (Kuespert and McAvoy, 1994). Therefore important operator actions could be isolated and their expertise could be captured without ad-hoc interviewing techniques (Kuespert and McAvoy, 1994). Thus not only qualitative insights can be gained from data analysis, but also numerical insights can be gained since conceptual models of these processes can be formulated from statistical/data driven techniques (Kuespert and McAvoy, 1994).

A data-driven approach could discover and identify modes within historical data. Knowledge of these modes can then be utilized to interpret, plan and execute operation more effectively. Abnormal conditions usually, however, do not manifest into a mode, but are rather addressed by supervisors prior (Liu and Chen, 2010). Abnormal conditions may therefore only make up a small fraction of the entire historical dataset. A data-driven state-based decision support system may therefore not effectively be able to identify the specific abnormal conditions, but could definitely assist with the various "normal" operating conditions.

2.1.3 The need for a data-driven approach

The data-driven approach to state-based decision support will address the following drawbacks of the fundamental modelling approach (Smarra, Jain and de Rubeis, 2018):

1. Many assumptions can be avoided since complex dynamics are captured within the data.

2. Since a data-driven approach works with sensor data directly, explicitly modelling internal states can usually be avoided.
3. Developed techniques are often deployable to various processes, therefore effort spent on the specific application is more valuable.

The goal of a data-driven approach is to learn from historical measurable data without modelling the physical details (Smarra, Jain and de Rubeis, 2018). Abnormal events will always occur, unseen within historic data, therefore the need for human supervisory staff remains critical. The dependency of modern control structures on humans remains high, therefore the development of any monitoring or decision support system should maximize the effectiveness of human-machine interactions.

A major drawback of modern data-driven approaches, however, is that the resulting models are often black box. Black box models are simply the functional relationships between system inputs and outputs, thus black box models are lumped together with parameter models. These parameters however do not have any physical significance when comparing them to process parameters such as heat and mass transfer coefficients (Zhang, 2010). If such models are used for state-based decision support, the reasoning for the suggested advisory is not clear. This is their major disadvantage and as a result, black box control designs are often not tolerated in the process industry; concerns of safety and robustness have blocked their introduction within control structures (Kuespert and McAvoy, 1994).

An emphasis should be made on the establishment of trust within the human supervisory control environment. Trust needs to be established between control operators, different shift teams, as well as humans and technology (Ashleigh and Stanton, 2001). Quality of interaction, understanding and confidence are key constructs that establish trust within the hierarchy in human supervisory control (Ashleigh and Stanton, 2001). To raise the level of trust within state-based decision support, systems need to be designed such that they respond in a “human-centered” manner (Ashleigh and Stanton, 2001).

Supervisors should not only be able to utilize the support system, but they should also be able to effectively make key design decisions during the data-driven model formulation. This would improve the interaction quality, understandability, and confidence within the control environment. This should result in raised levels of trust within control operators, different shifts, and human-machine systems. Ultimately it will increase the likelihood of actual implementation of a system developed in this manner. Data mining of historical process and supervisor action data could assist in the development of a state-based decision support system that is human-centered. Although, such a system would provide useful decision support, it would not nullify existing support solutions such as model predictive control or rule based control, but could instead be used in conjunction with these existing methodologies.

2.2 Data Mining and Knowledge Discovery

Data mining and knowledge discovery is performed with the main purpose of developing methodologies and tools to automate the data analysis process, such that useful knowledge is extracted from data that will ultimately allow for more effective decision making (Wang, 1999). Data mining is used in various industries and needs to be developed according to its intended purpose. Most developed systems, however, make use of various integrated techniques and approaches that condense data into knowledge

that can ultimately be leveraged for decision making (Wang, 1999). A decision support system should address process data characteristics that may pose a challenge to the data-driven model development.

2.2.1 Characteristics of Process Operational Data

Process industry data characteristics which may pose challenges are described as follows (Wang, 1999).

- Large volume: Automatic data logging produces massive amounts of data due to the large number of considered variables in process plants. This requires large amounts of computer memory and processing power.
- High dimensionality: A large number of correlated variables are considered, this makes it difficult to visualise data unless tools of dimension reduction are used.
- Noise: Noise introduced due to instrumentation faults or sensor noise, affect data pre-processing steps that need to be taken (Srinivasan, Viswanathan and Vedam, 2005).
- Dynamics: Many of the data processing tools are only able to deal with categorical values, these are however usually not obtained in process plants. Continuous-valued variables that have dynamic trends are the kind of values obtained from process plants.
- Sampling rate: Different variables have different sampling rates depending on the method of analysis such as on-line or laboratory analysis.
- Incomplete data: Some key variables may not be recorded.
- Complex interactions: Many process variables are interrelated, but certain tools require data to be independent.
- Redundant measurements: Multiple sensors may be used to measure the same variable.

It is important to be aware of the complexity of the data produced in the process industry and the challenges it poses. As a result, a number of techniques are required to prepare data for analysis, producing a multifunctional and integrated system (Wang, 1999).

2.2.2 Decision Support System Functions

Various forms of decision support systems exist within the context of continuous processes. Model development objectives should be set such that these functions are incorporated within the support system. These functions are required in all variants of support systems, to allow for system flexibility (Wang, 1999):

- Pattern discovery: This is a good starting point for data assimilation, in which data is grouped into clusters and analysed according to their similarities and dissimilarities.
- Link and dependency analysis: The link between performance metrics and the assimilated data is an important relationship that needs to be known, which will ultimately improve understanding of process behaviour and performance.
- Sequential pattern analysis: Aims at generating knowledge from the sequence of time series data.
- Trend and deviation analysis: Due to aging equipment, catalyst deactivation, and sensor drift previously identified patterns may have deviated from the original mode defining characteristics. An indication of when these variations occur could be of importance and will improve the integration of a state-based monitoring system within the control system (Xie and Shi, 2012).
- Summarising: Providing a compact description of assimilated data.

Within literature, some of the key decision support variants can be summarized into process monitoring and process optimization. Process monitoring has been determined to consist of four sequential tasks; fault detection, fault identification, fault diagnosis, and process recovery (Chiang and Russell, 2001; Quiñones-Grueiro, Prieto-Moreno and Verde, 2019). A process fault can be described as an "abnormal" process condition or event, for example resulting due to sensor failure (Chiang and Russell, 2001). Fault detection is determining whether the fault has occurred (Chiang and Russell, 2001), thus determining if the process is in a normal or abnormal process condition. Processes that follow after fault detection would be fault identification and diagnosis. Fault identification requires that the fault causing variables are identified, after which fault diagnosis requires that the exact cause is determined (Chiang and Russell, 2001). Only then can process recovery occur, which requires that procedures are performed that remove the fault's effects (Chiang and Russell, 2001).

Similarly, as discussed in 2.1.2.2, process optimization requires classification of process states (disturbance caused and operator-induced), the states' optimality, and determination of the procedures required to optimize sub-optimal conditions while monitoring the process constraints. Both process optimization and monitoring thus require the "discovery" of the normal operating conditions as well as the procedures required to optimize a process (process optimization) or return to normal operation (process monitoring). The similarities, as well as dissimilarities of the state-based decision support variants, can be seen in Figure 2.

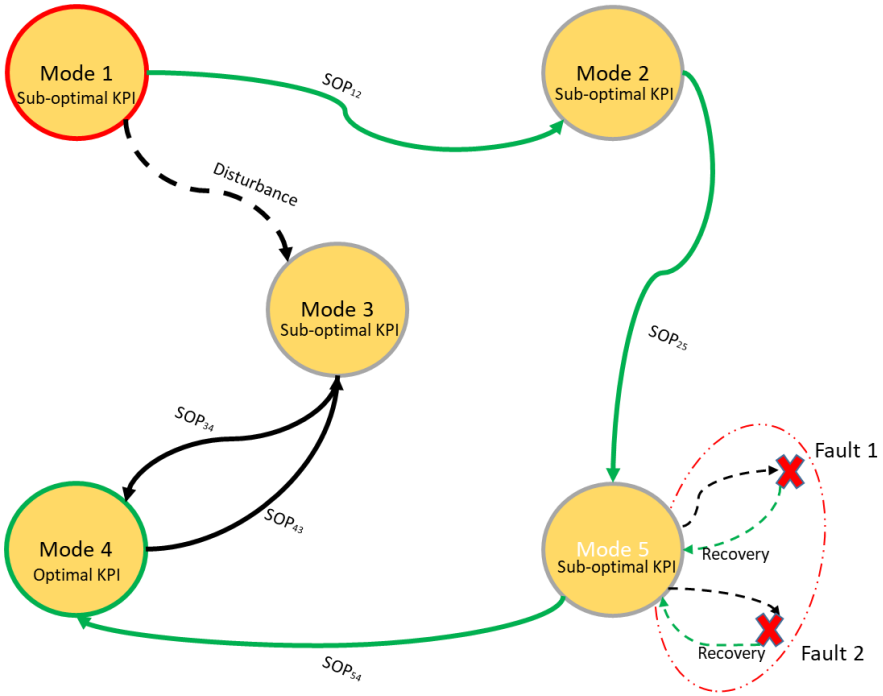


Figure 2: Diagram displaying state-based decision support system variants: process monitoring (enclosed in the red circle) and process optimization (not enclosed in the red circle)

From Figure 2 it should be seen that the modes or normal states of operation are accounted for in both process optimization and process monitoring. Abnormal states or faults are however typically not explicitly modeled, since processes do not manifest into faulty modes (data scarcity), as a result of supervisor intervention (Liu and Chen, 2010). Within the scope of this investigation, fault states are thus not considered. The state-based decision support system should therefore only assist with normal operation (or sub-optimal normal operation), addressing objectives set in 1.3.

2.3 Fault Detection and Diagnosis

What should be realized, is that a fault detection and diagnosis systems and the described decision support system (objectives described in 1.3) are very similar (process optimization), since they both require adherence to system support functions mentioned in 2.2.2. Thus, methods and literature from fault detection and diagnosis can be implemented to achieve the objectives set for this investigation.

2.3.1 Process Monitoring in the Past

Two types of variations can be found in process data; these variations may be either the result of a common cause or a special cause. Control systems are mostly able to remove special cause variations, however common cause variations such as sensor noise will always be present. Process monitoring should be able to distinguish the two variations within process data. Statistical theory thus plays a vital role in process monitoring schemes (Chiang and Russell, 2001).

The assumption of repeatable characteristics such as mean and variance within the same operating condition is required for the implementation of statistical theory. Thresholds can therefore be set, which allow for partial automation of the monitoring process. Historically Shewhart charts were employed to monitor key variables with the use of statistically determined thresholds (Wang, 1999). An example of such a chart can be seen in Figure 3.

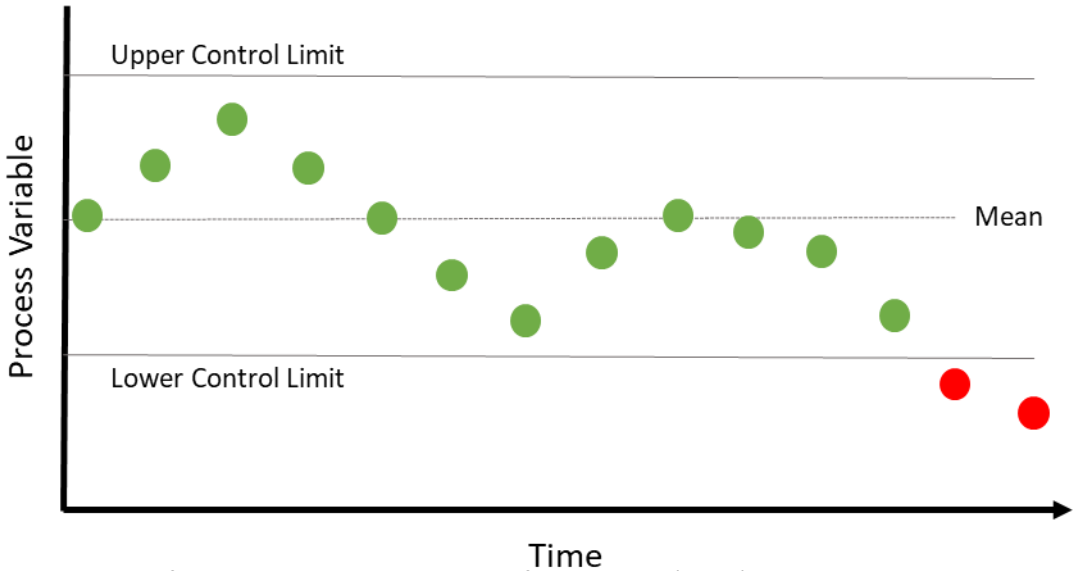


Figure 3: Diagram of a Shewhart chart redrawn from Wang (1999), displaying Control Limits. Green samples indicate common cause variations (for example sensor noise) and red samples indicate special cause variations (faults or process transitions)

This type of monitoring is univariate since it only compares a single variable against its upper and lower control limits. For diagnostic analysis, however, various Shewhart charts have to be monitored sequentially (Wang, 1999). Within the modern industrial context, this is not feasible, since massive amounts of multivariate data is collected continuously. This multivariate data may however also be correlated, therefore requires the need for the variables to be monitored simultaneously. Improved monitoring methods, therefore, had to be developed.

2.3.2 Principal Component Analysis

Multivariate statistical process control (MSPC) methods have widely been used in the process industry for the detection and diagnosis of abnormal conditions (AlGhazzawi and Lennox, 2008). Within the context of this investigation, such methods could assist with the discovery and identification of modes within data. In particular, principal component analysis (PCA) has demonstrated to be robust and effective for real-time monitoring in industry. The central idea behind PCA is to reduce the dimensionality of a dataset which consists of a large number of interrelated variables while retaining as much variation as possible (Jolliffe, 2002).

PCA does this by determining m new orthogonal vectors called loading vectors that describe the input dataset $\mathbf{X}$, which are ordered according to the amount of variance explained. A training set of n observations and m dimensions or process variables is contained in matrix form as described in Equation 2-1.

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \dots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \qquad 2\text{-}1$$

Prior to performing PCA, data is usually normalized. If there are large differences in the variance within variables, then the variables which vary with larger magnitudes will dominate the principal components (PCs) (Jolliffe, 2002). For example, temperature measurements may vary with far greater extents than concentration measurements, the variance in both may however be equally important. To address this issue PCA is usually performed on the z-scores of the input data ($\mathbf{X}$), which are determined as described by Equation 2-2 to 2-4.

$$\mu_j = \frac{\sum_{i=1}^{n} x_{ij}}{n} \qquad 2\text{-}2$$

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{n} (x_{ij} - \mu_j)^2}{n}} \qquad 2\text{-}3$$

$$\widehat{x_{ij}} = \frac{x_{ij} - \mu_j}{\sigma_j} \qquad 2\text{-}4$$

Here, μ_j and σ_j refer to the mean and standard deviation of variable j. The z-scores ($\widehat{x_{ij}}$) are then further analyzed in the matrix format as described by Equation 2-1. The loading vectors of the principal components are determined by solving the stationary points of the optimization problem shown in Equation 2-5 (Chiang and Russell, 2001).

$$\max_{\mathbf{v}\neq 0} \frac{\mathbf{v}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{v}}{\mathbf{v}^{\mathbf{T}}\mathbf{v}} \qquad 2\text{-}5$$

Where $\mathbf{v}\,\epsilon\,R^m$ and denotes the loading vector of a principal component (PC). The first PC is therefore given by the linear combination of m variables, as seen in Equation 2-6 (Wang, 1999).

$$\mathrm{y}_1 = \emptyset_{11}x_{i1} + \emptyset_{21}x_{i2} + \cdots + \emptyset_{m1}x_{im} \qquad 2\text{-}6$$

The loadings shown by ∅ are stored in the mentioned loading vector, as shown in Equation 2-7 for the loading vector of the first principal component y_1.

$$\mathbf{v_1} = \begin{bmatrix} \emptyset_{11} \\ \vdots \\ \emptyset_{m1} \end{bmatrix} \qquad 2\text{-}7$$

The first principal component, therefore, lies in the direction of maximum variance within $\boldsymbol{X}$, this direction is described by $\mathbf{v_1}$, such that $\mathbf{v'_1 v_1} = 1$. What should be clear is that the first principle component and all others are linear combinations of all m variables (constant coefficients or loadings), therefore PCA is a linear dimensionality reduction technique. The effectiveness of PCA is thus diminished when it is performed on non-linear data since it is not able to retain the data's variance as effectively within the reduced dimensional space.

The second principle component is once again determined by solving the optimization described in Equation 2-5, however with an additional constraint described by Equation 2-8 (Wang, 1999).

$$\mathbf{v'_2 v_1} = 0 \qquad 2\text{-}8$$

Thus Equation 2-8 denotes that the second principle component must be orthogonal to the first. The fact that the principle components are orthogonal provides various advantageous, which will be discussed later. This process is repeated such that m principle components are obtained, which are all orthogonal to each other.

The described optimization problem can conveniently be solved by means of eigenvector decomposition, which is described as follows. When $\boldsymbol{X}^{\mathrm{n\times m}}$ has a multivariate normal distribution, ie. $\mathbf{X}\sim N(0,\boldsymbol{\Sigma})$, the covariance matrix ($\mathbf{S}^{\mathrm{m\times m}}$) can be estimated with the use of Equation 2-9. Eigenvector decomposition of the covariance matrix is therefore described by Equation 2-9.

$$\mathbf{S} = \frac{1}{\mathrm{n}-1}\mathbf{X}^{\boldsymbol{T}}\mathbf{X} \qquad 2\text{-}9$$

$$\mathbf{S} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}} \qquad 2\text{-}10$$

Where $\mathbf{V}^{\mathrm{m}\times\mathrm{m}}$ denotes the matrix containing the eigenvectors or the loading vectors as its columns and $\boldsymbol{\Lambda}$ denotes the eigenvalue matrix, as seen in Equation 2-11 (Yu and Qin, 2008). The eigenvalue matrix contains the eigenvalues in order of decreasing magnitude, as described by Equation 2-12.

$$\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_m) \qquad 2\text{-}11$$

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0 \qquad 2\text{-}12$$

2.3.2.1 Dimensionality Reduction

By reducing the dimensionality of data, the number of independent parameters required to be estimated in further procedures are reduced, this may increase the bias of the estimate, but will also reduce the variance of the estimate (Chiang and Russell, 2001). This is known as the bias-variance tradeoff. Bias refers to approximating a complex real-life problem or dataset by a simpler model (James *et al.*, 2013), or in this case a reduced dimensional space. Variance here refers to the amount by which a classification/prediction (for example normal or abnormal operation) changes if a different training set was used (James *et al.*, 2013), thus influencing the repeatability of statistical process control. When the decrease in variance outweighs the increase in bias, dimensionality reduction results in better parameter estimates and could result in reduced misclassification (fault or normal operation). PCA however does not explicitly optimize for this.

Increased dimensionality for a fixed number of data patterns could lead to a sparse data space, therefore may result in misclassification (Wang, 1999). For example, clustering algorithms (discussed later) make use of similarity metrics such as the Euclidean distance to discriminate between different data clusters. As the number of variables considered increases the discrimination power of similarity metrics decreases (Thomas, Zhu and Romagnoli, 2018). As a result, more data is required to improve the accuracy of the parameter estimates at higher dimensions.

Reducing the dimensionality of data may however not only improve classification in subsequent steps but will also assist with data visualization. PCA is able to retain most of the variance of a dataset within a few components, this results in the removal of noise and captures the main structure within the high dimensional dataset. The greater the degree of correlation within the original variables, the fewer principle components are required to describe the input dataset (Wang, 1999). The structure retained by PCA can be useful for identifying variables causing or most affected by a fault (Chiang and Russell, 2001), thus assisting in summarization which is a key component of a state-based decision support system.

In certain cases PCA can reduce the dimensionality of data such that modes become visible within fewer dimensions (Aldrich and Auret, 2013). Modes can be identified as regions where the scores are concentrated in small ellipses (Srinivasan, Wang and Ho, 2004), the presence of transient states within process data however makes mode discovery more difficult.

Various techniques have been developed to determine the reduction order of the original dataset, however no dominant technique exists (Chiang and Russell, 2001). For example one of the techniques often employed to determine the reduction order is cumulative variance explained.

The fraction of variance explained by a specific principal component (θ_{VE_j}) can be determined using Equation 2-13. Where subscript j denotes the investigated principal component. Using the phenomena described in Equation 2-13, Equation 2-14 can then be utilized to select the extent of dimensionality reduction.

$$\theta_{VE_j} = \frac{\lambda_j}{\sum_{i=1}^{m} \lambda_i} \qquad 2\text{-}13$$

$$k \leftarrow \min(k) \text{ with } \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{i=1}^{m} \lambda_i} > \theta_{CVE} \qquad 2\text{-}14$$

Here k principal components are selected such that the cumulative variance explained threshold (θ_{CVE}) is exceeded. Thus the dimension of the original dataset is reduced to the latent space $\mathbf{P}^{\mathrm{m\times k}}$, as long as $k < m$.

2.3.2.2 Fault Detection using PCA

The correlated high dimensional input data ($x_i \in \mathbb{R}^M$) can thus be projected into an uncorrelated lower dimensional latent or principal component space ($z_i \in \mathbb{R}^k$), as described by Equation 2-15.

$$\mathbf{Z} = \mathbf{XP} \qquad 2\text{-}15$$

Here $\mathbf{Z}^{\mathrm{n\times k}}$ denotes the score matrix, which contains the input data in the latent or principal component space. $\mathbf{P}$ denotes the PC loading matrix, where the various loading vectors ($\mathbf{v}$) are assigned to the columns of the matrix. Hotelling's T^2 is a popular MSPC index that allows for the measure of variability within the normal latent space. It is an indication of how far a score is from the multivariate mean of the data, i.e the intersection of the principal components (Slišković, Grbić and Hocenski, 2012). If a single score is taken from the score matrix $\mathbf{Z}$ it will have the row vector format shown in the Equation 2-16. Hottelling's T^2 for observation i can therefore be calculated using Equation 2-17.

$$\mathbf{z}_i = [z_{i1}, z_{i2}, \dots, z_{ik}] \qquad 2\text{-}16$$

$$\mathrm{T}_\mathrm{i}^2 = \sum_{j=1}^{k} \left(\frac{z_{ij}}{\lambda_j}\right)^2 \qquad 2\text{-}17$$

Figure 4 gives an illustration of how the Hotelling's T^2 is obtained. Firstly described by step one (arrow 1) PCA is performed on the multivariate normal dataset. Therefore, the data is rotated such that the scores correspond to the principal components, this is achieved with Equation 2-15. The scores are then scaled by dividing them with their corresponding eigenvalue, resulting in scores with unit variance in all latent

directions, as seen in Figure 4 step two. This procedure allows a scalar threshold to characterize the variability of a high dimensional input dataset (Chiang and Russell, 2001).

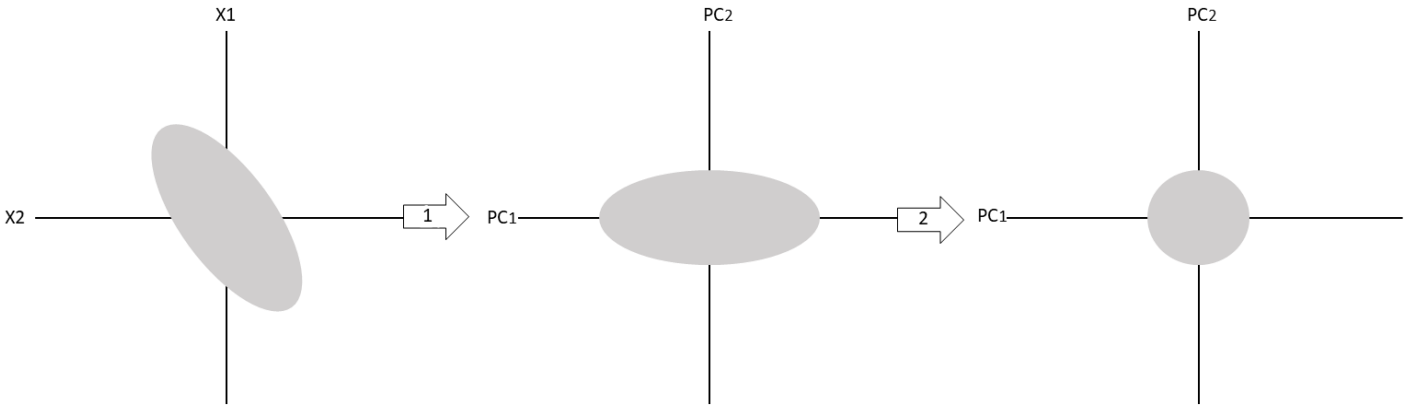


Figure 4: Illustrating the conversion of the covariance such that the T^2 statistic threshold can be determined (redrawn from Chiang and Russel (2001))

A threshold can then be determined to automate the procedure of fault detection. A threshold for the Hotelling's T^2 statistic is determined by applying the χ^2 probability distribution, assuming that the samples are randomly sampled from a multivariate normal distribution, as mentioned earlier. However, since the actual covariance of the data is not known rather estimated (Equation 2-9), a threshold can be determined using Equation 2-18 (Chiang and Russell, 2001), here $F_{\propto}(k, n-k)$ denotes the F-distribution.

$$T_{\propto}^2 = \frac{k(n-1)(n+1)}{n(n-k)} F_{\propto}(k, n-k) \qquad 2\text{-}18$$

At a given significance level, as the correlation between variables increases, the elliptical confidence region elongates, resulting in less conservative confidence boundaries compared to the univariate case (Chiang and Russell, 2001). The differences in conservatism can be seen in Figure 5.

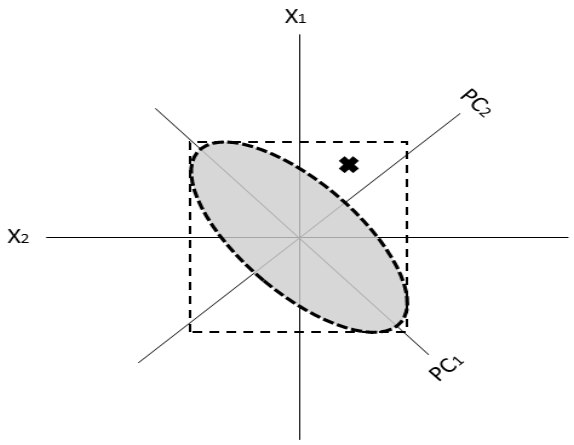


Figure 5: Hotelling's T^2 confidence interval compared to the univariate confidence interval indicated by the box (Redrawn from Chiang and Russel (2001))

Figure 5 also shows an anomaly or fault which would not be detected by either of the univariate cases, but would be detected if the Hotelling methodology was used. Referring back to the "curse of dimensionality", due to the nature of Equation 2-17, when principal components describe a small variance (small eigenvalues), it may lead to erratic behavior in the T^2 statistic. This therefore supports the need for dimensionality reduction, since it results in more robust fault detection (Chiang and Russell, 2001).

Another statistic is therefore required to account for variations occurring in directions where the eigenvalues are low. Thus the Q statistic is incorporated to monitor deviations from the high dimensional input not captured by the latent/PC space. The Q statistic can therefore describe how well the latent space captures the variance within the high dimensional space. Therefore, the scores need to be projected back into the high dimensional space, as described by Equation 2-19. Where $\mathbf{E}^{\mathrm{n}\times\mathrm{m}}$ denotes the residual matrix, which contains the residuals as described by Equation 2-20.

$$\mathbf{E} = \mathbf{X} - \mathbf{Z}\mathbf{P}^{\mathrm{T}} \qquad 2\text{-}19$$

$$\mathbf{E} = \begin{bmatrix} \mathrm{r}_{11} & \cdots & \mathrm{r}_{1m} \\ \vdots & \ddots & \vdots \\ \mathrm{r}_{n1} & \cdots & \mathrm{r}_{nm} \end{bmatrix} \qquad 2\text{-}20$$

The Q statistic for a specific observation i can therefore be calculated by Equation 2-21. The Q statistic can, therefore, be seen as the "goodness of fit" or squared prediction error (SPE), where $\mathbf{r_i} = [\,\mathrm{r_{i1}}, \ldots, r_{im}]$. The SPE threshold ($Q_\propto$) is determined via Equation 2-22, where $\theta_i = \sum_{j=k+1}^{m} \lambda_j^{2\,i}$, $h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}$, and c_α is the normal deviate corresponding to the $(1-\alpha)$ percentile. Violations Hotelling's T^2 threshold suggest shifts in data mean and covariance structure, whereas violations of the SPE threshold suggest a break in the expected correlation within the data (Chiang and Russell, 2001; Aldrich and Auret, 2013).

$$Q_i = \mathbf{r_i}\mathbf{r_i^T} \qquad 2\text{-}21$$

$$Q_\propto = \theta_1 \left[\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}} \qquad 2\text{-}22$$

After a fault has been detected both in the residual or latent space, various procedures of fault identification (identify the fault causing variables) and diagnosis (root cause determination) can be utilized (Chiang and Russell, 2001; Aldrich and Auret, 2013). For example, contribution plots can be utilized to assist in the identification of faults, or in the context of this investigation mode shifts. A diagram of a contribution plot can be seen in Figure 6. Here the residuals of specific variables (r) are compared against each other. An issue however with contribution plots is that the contributions do not assist in the discrimination of the effect and the cause of a fault/transition (Aldrich and Auret, 2013). Thus, variables that contribute to a significant magnitude to the overall contribution may be the fault causing variables or they may just be symptoms of the fault (Aldrich and Auret, 2013).

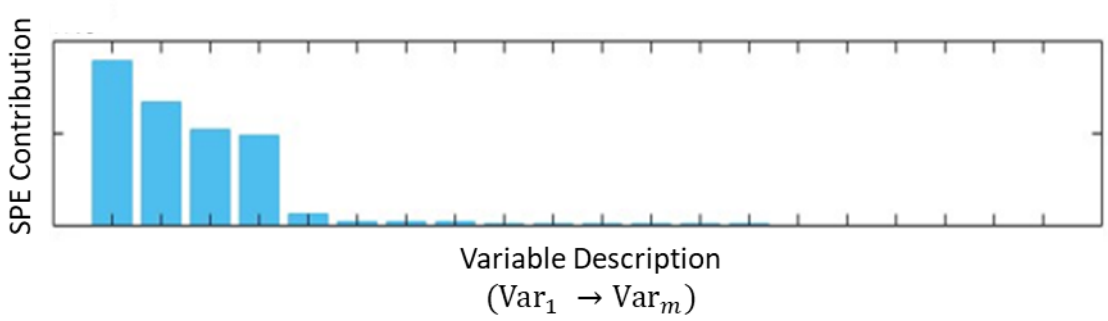


Figure 6: Diagram of a contribution plot indicating variables that may be either the symptoms or causes of a fault/transition

Expert knowledge of process topology as well as knowledge of time lags within complex processes become crucial when determining the root cause of a fault (Aldrich and Auret, 2013). Only once the root cause of a fault has been determined can process recovery occur. Methods of causality analysis have been developed to automate the determination of cause-effect variable pairs (Aldrich and Auret, 2013), these methods are, however, outside the scope of this investigation. Rather attempts at leveraging supervisory control action datasets will be made (datasets discussed in 2.1.2.4).

2.3.2.3 PCA Limitations

Various factors have been determined to affect the viability of PCA for process monitoring. Firstly, process data tends to be autocorrelated or serially correlated as a result of for example recycle streams and high sampling rates (Chiang and Russell, 2001). Controllers are usually unable to effectively deal with such variations, thus these variations exist within the data. Since PCA only seeks to retain maximum variance on latent variables linearly dependent on the input variables, standard PCA won't be able to capture the autocorrelation occurring within process data. If however enough data is available to represent the normal process variations, the effectiveness of PCA is not compromised (Chiang and Russell, 2001). An extension of PCA, called Dynamic PCA is more robust to the impact of autocorrelation since the input space contains additional lagged variables of the original variables themselves.

Secondly, process data is mostly not linearly related, as a result, PCA is not able to effectively retain the variance occurring within data. To attend to this problem PCA has been extended to Kernel PCA in many instances, where non-linear variables are mapped into the input data and then the PCs are determined (Zhang, Li and Teng, 2012). This however results in increased complexity, subsequently steps of fault identification and diagnosis are more difficult. Additionally, a major problem that monitoring applications have to deal with is process drift, in other words, process data is not time-invariant. Various reasons exist why a process may drift such as aging equipment or catalyst deactivation (Xie and Shi, 2012). To address this issue adaptive extensions of PCA were developed such as recursive PCA or moving window PCA. These extensions require that the PCA "model" is continually updated with new data in various forms, thus resulting in reduced misclassification.

Lastly, PCA requires the assumption that multivariate data obeys a unimodal Gaussian distribution (linear). However, due to the fact that processes are run in an agile (operate at various states) manner, this assumption is largely not obeyed. As a result of drift and the mentioned transitions to various modes, process data cannot be described by a single Gaussian. Rather process data is non-linear and multimodal as a result of mode changes, the combination of which makes standard PCA and its mentioned extensions invalid for MSPC.

To account for the above-mentioned problem multiple mode PCA has been implemented (Ha *et al.*, 2017). Therefore a different set of principle components (and standardization parameters) are used to monitor the different modes, otherwise known as local principal components. Unfortunately, data-driven fault detection methods such as the multimode PCA are applied in a supervised manner, such that data is labelled into classes manually. This requires that experienced supervisory staff label historic data into different classes such as normal operation or faults. However, with the complexity and extensive amount of process data, this does not occur often in an industrial setting (Thomas, Zhu and Romagnoli, 2018). As a result additional unsupervised learning strategies are crucial for system support functions, mainly aimed at achieving pattern discovery, mentioned in 2.2.2.

What should however be realized is that PCA remains useful due to its reduced design efforts and simplicity of its fundamental premise. PCA has special relevance when process data has a multivariate normal distribution, its usefulness should however not be detracted when data has other forms (Jolliffe, 2002). Due to these reasons PCA is predominantly applied in literature as well as industry for dimensionality reduction and MSPC (its extensions) (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019). PCA could be utilized to reduce the dimensionality of the process data considered within this investigations, but its extensions or different approaches would have to be utilized to account for its limitations.

2.4 Data Clustering

Within the context of this investigation, various process modes need to be discovered from historical process data, only then can the switching procedures between the modes be determined. Unsupervised learning allows for patterns to be deduced from data where the response is unknown. An additional procedure of clustering is therefore required, which allows for the grouping of data into classes (ie. modes) such that patterns within a class are similar and distinct from other classes (Wang, 1999).

2.4.1 Most Popular Clustering Techniques for Multi-modal Processes

No universal data-driven algorithm works well under all conditions, as a result, no dominant technique exists (Chiang and Russell, 2001). This is known as the “No Free Lunch Theorem” (Wolpert and Macready, 1995). Various clustering techniques have been implemented for data mining and monitoring of multimodal processes. A summary of the most popular techniques is provided as follows (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019):

- Mixture models: This technique fits various density functions to a population that contains several local populations using the iterative procedure of expectation maximisation. As described

the result is a probabilistic model, therefore the samples are not directly clustered, however with the incorporation of a threshold samples can be clustered. Mixture models (MM) have been determined to be the most popular clustering technique for multimodal processes. MM methods do however have certain issues, such as the fact that the number of modes or clusters needs to be known prior. Robust extensions of the algorithm are also required to deal with noise/outliers since all samples are considered during convergence of EM (Choi, Park and Lee, 2004; Yu, 2011).

- K-means: K initial centroids are chosen, then the Euclidean distance of the remaining samples in the dataset are calculated to the chosen centroids. The procedure is then iterated seeking the minimization of the sum of squared errors to the mean of the cluster (Thomas, Zhu and Romagnoli, 2018). Thus, the number of modes/clusters must be known prior to the analysis. In the standard algorithm cluster assignments of the samples are also hard, meaning that a sample belongs to one cluster only. The cluster shapes (covariance) are also fixed to be spherical since the Euclidean distance metric is used. The algorithm may also converge to local minima as a result of the choice of poor initialization parameters (initial centroids), mixture models may also experience the same issue.
- Fuzzy C-means: This is an extension of the K-means clustering technique, where sample points have soft cluster assignments. As a result samples have membership values to various clusters. This technique can more effectively deal with outliers, however, does not fare well when transition data is included within the dataset (Wang, Wang and Wang, 2013).
- Window-based: Since process data is mostly time-series, sequential analysis of the data may provide various benefits. Various window-based approaches have been developed, such as sequential clustering (Srinivasan, Wang and Ho, 2004). The main advantage of implementing such techniques is that the number of modes could be determined automatically. Most methods are also able to distinguish between transitions and modes. This is an important attribute since processes are not able to settle to transient states. These techniques are however heavily dependent on the parameter of window sample size. Determining a similarity measure between windows is also not an easy task. As a result, setting the parameters of such techniques is difficult (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019).

Therefore the various mentioned unsupervised methods have certain drawbacks and benefits. Most techniques are not able to deal with transition data or outliers and others may result in complex monitoring models. Implementation of a single method will result in ineffective knowledge discovery and thus supports the need for an integrated approach that makes use of various techniques. This will allow the benefits of the various techniques to be maximized and their inherent drawbacks to be overcome.

2.4.2 Gaussian Mixture Models

Gaussian mixture models (GMM) are a structure of mixture models that are able to model data that consists of multiple multivariate Gaussian distributions. GMMs are therefore well suited for representing process data, since each operating mode can be described by a local Gaussian. As a result, GMMs have recently been extensively used for multimodal process monitoring. They could therefore also be used for the discovery and identification modes within process data.

2.4.2.1 Fundamentals

The probability density function of a multimodal multivariate ($\mathbf{x} \epsilon R^m$) is described by Equation 2-23 (Yu and Qin, 2008).

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{i=1}^{K} w_i g(\mathbf{x}|\boldsymbol{\theta}_i) \qquad 2\text{-}23$$

Here K describes the number of modes or Gaussian components, w_i the weight (or prior probability) of the ith component (mode) and $\boldsymbol{\theta}_i = \{\mu_i, \sum_i\}$ contains the local Gaussian model parameters. The ith component or mode density function is described by Equation 2-24.

$$g(x|\boldsymbol{\theta}_{\mathrm{i}}) = \frac{1}{(2\pi)^{\frac{m}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathrm{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathrm{x} - \boldsymbol{\mu}_{\mathrm{i}})\right] \qquad 2\text{-}24$$

Since Gaussian model parameters are unknown, they need to be estimated. Various methods have been implemented to estimate these parameters, the expectation maximization (EM) algorithm has however been the method used most extensively in practice (Yu and Qin, 2008). The EM algorithm is implemented iteratively by repeating an expectation step (E-step) and a maximization step (M-step) until a convergence criterion of the log-likelihood function is satisfied. Given a training dataset $\mathbf{X}$, as described by Equation 2-1 and an initial estimate of the Gaussian model parameters, described by Equation 2-25, the EM algorithm can be performed (Yu and Qin, 2008).

$$\boldsymbol{\Theta}^{(0)} = \{\left\{w_1^{(0)}, \boldsymbol{\mu}_1^{(0)}, \boldsymbol{\Sigma}_1^{(0)}\right\}, \ldots, \left\{w_K^{(0)}, \boldsymbol{\mu}_K^{(0)}, \boldsymbol{\Sigma}_K^{(0)}\right\}\} \qquad 2\text{-}25$$

The E-step and M-step are performed iteratively as follows:

- E-Step

In the E-step the posterior probability of the jth sample belonging to a certain component is determined by means of Bayes' theorem as described in Equation 2-26 (Choi, Park and Lee, 2004).

$$P^{(\mathrm{s})}(C_c|\mathbf{x}_j) = \frac{w_c^{(s)} g(\mathbf{x}_j|\boldsymbol{\mu}_c^{(s)}, \boldsymbol{\Sigma}_c^{(s)})}{\sum_{i=1}^{K} w_i^{(s)} g(\mathbf{x}_j|\boldsymbol{\mu}_i^{(s)}, \boldsymbol{\Sigma}_i^{(s)})} \qquad 2\text{-}26$$

Here C_c denotes the cth component, x_j the jth sample and $P^{(s)}(C_c|x_j)$ describes the posterior probability at the sth iteration.

- M-Step

Then in the M-Step the $\boldsymbol{\Theta}^{(s+1)}$ parameters are determined as follows.

$$\boldsymbol{\mu}_c^{(\mathrm{s+1})} = \frac{\sum_{j=1}^{n} P^{(s)}(C_c|\mathbf{x}_j)\mathbf{x}_j}{\sum_{j=1}^{n} P^{(s)}(C_c|\mathbf{x}_j)} \qquad 2\text{-}27$$

$$\boldsymbol{\Sigma}_c^{(s+1)} = \frac{\sum_{j=1}^{n} P^{(s)}\left(C_c|\mathbf{x}_j\right)\left(\mathbf{x}_j - \boldsymbol{\mu}_c^{(s+1)}\right)\left(\mathbf{x}_j - \boldsymbol{\mu}_c^{(s+1)}\right)^T}{\sum_{j=1}^{n} P^{(s)}\left(C_c|\mathbf{x}_j\right)} \quad 2\text{-}28$$

$$w_c^{(s+1)} = \frac{\sum_{j=1}^{n} P^{(s)}\left(C_c|\mathbf{x}_j\right)}{n} \quad 2\text{-}29$$

The E-step therefore determines the posterior probability given all the parameters $\boldsymbol{\Theta}$. The M-step then involves finding all parameters given the posterior probability (updates). The iterative nature of the EM algorithm allows the calculation to then converge to a stable solution, which can be described as the maximum likelihood solution.

The log likelihood can be calculated by Equation 2-30, the Log is used to improve intelligibility (likelihood becomes more discernable from zero) (Yu, 2011). The EM algorithm has converged when Equation 2-31 is satisfied.

$$\log L(\boldsymbol{\Theta}) = \sum_{j=1}^{n} \log\left(\sum_{i=1}^{K} w_i g(\mathbf{x}|\boldsymbol{\theta}_i)\right) \quad 2\text{-}30$$

$$\widehat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} (\log L(\boldsymbol{\Theta})) \quad 2\text{-}31$$

Finally, the GMM is able to capture the linear relationships among its variables in $\boldsymbol{\Sigma}_c$ as well as the operating point of the mode ($\boldsymbol{\mu}_c$), making it an ideal model for multimodal processes (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019).

2.4.2.2 Multimodal Fault Detection using GMM

Once a GMM is fit to training data certain measures can be implemented for the detection of anomalies or in other words determine if a sample belongs to any of the modes within the GMM. The negative likelihood probability (NLLP) is a measure that has been implemented for this purpose and can be obtained using Equation 2-32 (Yu, 2011).

$$NLLP = -\log p(\mathbf{x}|\boldsymbol{\theta}) \quad 2\text{-}32$$

The negative log of the likelihood is obtained here to make the measure more discernable from zero. Therefore if the NLLP is smaller than a set threshold, the process is deemed to be from a distribution contained within the GMM. NLLP thus is a global identification measure, indicating only whether the process is in a normal condition or in a faulty (mode not contained within the GMM) condition and does not specify what the current mode of a sample is. Since the training data is not normally distributed, unlike unimodal fault detection, control limits cannot be determined directly from particular approximate distributions (Yu, 2011). To overcome this deficiency density of mixture modelling techniques have been used to estimate an overall confidence boundary around multiple clusters. However since this method

requires Monte Carlo simulation-based random sampling, it is undesirable for industrial applications (Yu and Qin, 2008).

Another measure known as the Mahalanobis distance has been implemented as well, which is able to quantify the deviation degree of a sample from the normal process state space (Yu, 2011). Unlike NLLP, this measure is however implemented in a local manner. This is due to the fact that the best matching component (BMC) or mode is calculated, which is the component with the minimum Mahalanobis distance. This distance metric is exactly the same as the Hottelling's T^2, therefore the metric is also described in Figure 4. The Mahalonobis distance can be determined by Equation 2-33.

$$D_{mahal} = (\mathbf{x} - \boldsymbol{\mu}_{BMC})\boldsymbol{\Sigma}_{\mathrm{BMC}}^{-1}(\mathbf{x} - \boldsymbol{\mu}_{BMC})^T \qquad 2\text{-}33$$

If D_{mahal} is smaller than a determined threshold, the process is deemed to be normal. What should also be realized is that if the metric is below the threshold, it is an indication that the process is currently in that specific mode (BMC), therefore can be viewed as a local monitoring method. This method can however also be used in the global monitoring context, here only D_{mahal} would be monitored, but would have a variable control limit depending on the BMC. If all components within the GMM had the same covariance structure, a single control limit could be implemented, this is however usually not the case. For example, the 99th percentile of the D_{mahal} for the specific BMC can be used as the control limit for the specific mode. What should however also be realized is that when the BMC is determined by means of Mahalanobis distance, the prior or weight (w_c) of the respective clusters is not taken into account. A different method thus exists, where the BMC is determined based on the highest posterior probability, described by Equation 2-26. This method of BMC determination may however be prone to issues resulting from data imbalance.

2.4.2.3 Limitations of GMM

Similar to PCA, GMM has certain limitations that make effective monitoring difficult. Firstly, GMMs suffer from the curse of dimensionality. When the data dimension is large, data may become sparse and as a result, the determination of a GMM is difficult or impossible (Choi, Park and Lee, 2004). The variance bias tradeoff also becomes an issue when the dimensionality is high, since the number of parameters that need to be fit increases by $2K(m+1)$ when additional variables are added. Similarly, the number of parameters increases by $m + m^2$ when additional components or modes are included. Certain constraints can be implemented to reduce the number of parameters that need to be estimated, such as constraining the shape of the covariance matrix. For example, only diagonal covariance matrixes can be fit to the data. Thus it is assumed that the variables are uncorrelated and therefore result in fewer parameters that need to be estimated. Another constraint that can be set is to use a single covariance structure for all components or modes, thus reducing the number of parameters that need to be estimated. Implementing these constraints may however not always be viable and may lead to poor monitoring performance. Determining GMMs from the reduced dimensional space obtained from PCA may be a viable solution to reduce the number of parameters that need to be estimated (Choi, Park and Lee, 2004). Constraining the covariance to be diagonal may also be more valid in the latent/PC space since all latent dimensions are uncorrelated (Addo, 2019). However since PCA assumes data to originate

from a unimodal distribution, it does not guarantee that the covariance of the modes occurring within the latent space are uncorrelated.

Another key issue with GMMs is that the number of components (modes) is required to be known a priori. These parameters are however usually not known. As the number of components increases, it is inevitable for the likelihood to increase, however, this will also increase the number of parameters that need to be estimated. The frequency of Type I errors will most likely increase as result, reducing the robustness of the determined models. On the contrary, if the number of components decreases Type I errors may be suppressed but will be accompanied with an increased frequency of Type 2 errors (Choi, Park and Lee, 2004). Various deterministic methods of determining the number of the optimal number components have thus been developed. These methods attempt to maximize the log-likelihood, but introduce a penalty for the number of parameters required. Some examples of these methods are the Akaike information criterion (AIC), minimum description length (MDL), Figueiredo and Jain (FJ) algorithm ,and the Bayesian inference criterion (BIC) (Yu, 2011). BIC shows the tradeoff between a good fit quality and the number of parameters required to obtain the fit, described in Equation 2-34 (Schwarz, 1978). The GMM configuration with a minimum BIC value is then chosen to model the data.

$$BIC = 2\log L(\boldsymbol{\Theta}) - n_p \log(n) \qquad 2\text{-}34$$

Even if a precise number of Gaussian components is known a priori from process knowledge, the EM algorithm may converge to local maxima. The GMM will therefore fail to describe the data effectively, resulting in both an increased frequency of Type 1 and Type 2 errors. The initial parameter estimates set (Θ^0) are therefore required to represent the actual components as accurately as possible to reduce the chances of the EM algorithm converging to local maxima. To address this issue, the K-means clustering algorithm has often been implemented to determine the initialization parameters due to its relatively good performance and simplicity (Yu, 2011).

Lastly, transition data and outliers found within process data reduce the effectiveness of GMM. As described in 1.2, transient states differ from modes or quasi steady states, and as a result, cannot be described by the same component within the GMM. However since all samples are considered within the EM algorithm, it is inevitable that transient and outlier data will affect the convergence of parameter optimization (Liu and Chen, 2010). Expert knowledge is therefore required to determine which data is adequate for analysis, however, due to the complexity of process data, this is often not applicable in the industrial context (Thomas, Zhu and Romagnoli, 2018).

2.5 Stationarity analysis

Since process data is mostly unlabeled, the number of modes is not known, which is a key parameter required by most clustering algorithms for effective characterization of modes (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019). As mentioned decision strategies to determine the number of modes have been developed, using Bayesian information criterion (Thissen, Swierenga and De Weijer, 2005) and others, however few methods are able to deal with multimode process data containing transitions (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019). Visual inspection of the latent space of complex

processes to determine the number of modes a priori is mostly also not possible, since modes may be masked by transient data or due to the high dimensionality of the data. In rare cases where the number of modes is known before clustering, the inclusion of transient data within the analyses could still skew mode characterization during clustering. Transient data may form a considerable fraction of the entire multimodal dataset (Srinivasan, Viswanathan and Vedam, 2005). The importance of removing transient data prior to analysis increases when the process in consideration experiences more transient states.

Window-based methods have been used to tackle some of these mentioned problems, by analyzing process time series data sequentially. The main advantage of an analysis in this fashion, is that the number of modes can be determined automatically (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019). This however requires that the transitional states are distinguished from quasi steady states. Stationarity analysis or steady state detection (SSD) can be used for this purpose, thus avoiding making simplifying assumptions of process data.

Many SSD techniques apply the student t-test on subsets of the standardized residuals of the time series signals (Kelly and Hedengren, 2013). Attempts have been made to allow for drift within the analysis by assuming process data can be described as a random walk with drift, described by the Equation 2-35 (Kelly and Hedengren, 2013):

$$x_t = \mu + mt + \sigma\varepsilon_t \qquad 2\text{-}35$$

where mt describes the drift, μ the mean, ε_t a random variable with a standard normal distribution, and σ the standard deviation of the random variable. The technique dealt with the multivariate property of process data by choosing a set of key variables and performing the described method on them separately. The tuning of these steady state algorithms is difficult and optimal parameter selection is an open problem (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019). The high dimensionality of industrial data as well as topology/time lags exacerbates the problem since the selection of key variables to analyze for SSD is difficult. Feature selection is therefore an important consideration to ensure selected variables are uncorrelated (Wang, 1999). Kelly and Hedengren (2013) proposed a new method for SSD with promising results, but without addressing the issues of high-dimensional data.

The utilized algorithm is based on that by Kelly and Hedengren (2013), which will be referred to as threshold SSD. The original algorithm is window-based, therefore all calculations are performed over a predefined time period containing n samples of the variable x_t. The drift m, mean μ and variance σ^2 (see Equation 2-35) are sequentially determined, as shown in Equations 2-36 to 2-38.

$$m = \frac{1}{\mathrm{n}}\sum_{t=2}^{n}\frac{x_t - x_{t-1}}{Sampling\ Frequency} \qquad 2\text{-}36$$

$$\mu = \frac{1}{\mathrm{n}}\sum_{t=1}^{n}(x_t - mt) \qquad 2\text{-}37$$

$$\sigma^2 = \frac{1}{\mathrm{n}-2}\sum_{t=1}^{n}(x_t - mt - \mu)^2 \qquad 2\text{-}38$$

A student t-test is used to evaluate the null hypothesis (Equation 2-39) at a confidence level α (corresponding to a critical test statistic t_{crit}) for each of the n samples in the window:

$$\text{If } |x_t - \mu| \leq t_{crit}\sigma \text{ then } y_t = 1 \text{ else } y_t = 0 \qquad 2\text{-}39$$

This algorithm is able to perform SSD for the multivariate case, by assuming that a process is only at steady state if all M key variables are steady, as expressed by Equation 2-40 (Brown and Rhinehart, 2000).

$$Y_{t,system} = \prod_{j=1}^{M} y_{t,j} \qquad 2\text{-}40$$

The state of the system at time t (steady or transient) is determined by comparing the fraction of samples which reject the null hypothesis to a predefined threshold θ_{SS} (Equation *2-41*):

$$\text{If } \left(\frac{1}{N}\sum_{t=1}^{N} Y_{t,system}\right) \geq \theta_{ss} \text{ then window is steady; else window is transient} \qquad 2\text{-}41$$

2.6 Recent Developments in Research

In the previous sections an overview of the various continuous process data characteristics, state-based support system functions, fundamentals of normal operating mode discovery/identification, and the limitations of popular approaches is given. Here, a more in-depth overview of the recent advances in process monitoring/optimization is given.

2.6.1 Process Monitoring

Multimode continuous process monitoring has been determined to require three main tasks, the characterization of multimode data via clustering, characterizing fault data if faults have been labelled, and finally to develop a supervised monitoring scheme that is able to distinguish modes from faults (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019). Thus as mentioned in 2.3, the goals of process monitoring align with those of a state-based decision support system for process optimization (objectives described in 1.3), mainly concerning the discovery and identification of modes within complex process data. As a result, the recent advances within process monitoring can be utilized.

The technique most widely used for process monitoring due to its reliability and simplicity is PCA. However, as a result of the unimodal limitation of PCA, extensions had to be developed. Zhao et al., (2004) developed a multiple PCA model scheme that is able to deal with the multimodality of process data. By utilizing a metric based on the average of the cosines of the angles between pairs of PCs, the similarity of these latent spaces can be determined. Training, therefore, requires the iterative comparison of this angle/metric with a threshold. Once the similarity measure of all model angles is above the threshold, the models for the different modes have been determined. Hotelling's T^2 and SPE are then used to monitor online data, such that if the statistics are below the control limits for any of the determined models, the process is in a normal operating condition. Similarly, Ha *et al.* (2017) first clustered data within the principle component space using K-means and then used local PCA models to monitor the various modes.

However, as a result of the limitations mentioned in 2.3.2.3, alternative methods of process monitoring have been developed. Recently, Hidden Markov Models (HMMs) have been implemented to monitor multimodal processes. HMM is a viable technique due to its ability to not only model the multimodality but also capture the mode shifting probabilities of process data (Afzal, Tan and Chen, 2017).

Alternatively, GMMs have also been widely applied to monitor multimodal processes. Choi et al., (2004) implemented GMM on the principle component space to monitor processes. With the use of linear discriminate analysis (LDA) the dimensionality of fault data is reduced, such that the within fault variance is minimized and the inter fault variance is maximized. It should however be noted that LDA is a supervised learning technique, thus fault data has to be divided into classes or labelled prior to analysis. In industry, this is rarely the case (Thomas, Zhu and Romagnoli, 2018). GMM is then again applied to this latent space for fault identification. Liu and Chen (2010) also made use of PCA and GMM to monitor process data by systematically extracting operating modes. This is achieved by estimating reasonable initialization parameters for the EM algorithm by means of kernel density estimation (KDE). Similarly, Yu (2011) made use of the FJ algorithm to determine the number of components for the GMM fit to data obtained from a semiconductor process.

PCA and its extensions form a crucial aspect of most techniques implemented, PCA however only considers the mean and covariance of the data and therefore only preserves the global variance. PCA, therefore, lacks the ability to extract local intrinsic information from the data that may be crucial in the analysis (Yu, 2016). As a result, the use of PCA within academia has been diminishing over recent times (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019). Techniques such as locality preserving projections (LPP) and manifold learning have thus become more popular since they are capable of preserving local variance within the data. Yu (2016) made use of LPP in conjunction with GMM for monitoring process data, which was more capable of handling complicated distributions.

Within literature, however, certain simplifying assumptions about simulated process data are often made. Assumptions are often formulated (within a process model/simulation) to demonstrate the feasibility of the contributions (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019). A key assumption often made is that process data does not contain transitions and as a result, limits the flexibility of many

approaches mentioned. Process monitoring (fault detection and diagnosis) literature is therefore more concerned with developing a supervised monitoring scheme that is able to distinguish modes from faults. However, the exploration and characterization of data remain an important task, such as determining the number of modes existing within the process data or identifying the starting and ending times of transition states. These are issues not completely solved yet (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019).

2.6.2 Optimality Assessment

Literature most relevant to the objectives of this investigation (described in 1.3) has been dubbed "optimality assessment". Although various contributions have been made to address this field, research is still in its infancy. As with process monitoring, the objectives align (more so) with the described state-based decision support system.

Sebzalli and Wang (2001) made use of PCA and fuzzy clustering to discover operational strategies for rapid product changeover from process data of a fluid catalytic cracking process. It was determined that the various clusters found within the data correspond to the different grades of product produced. The analysis however did not consider the online implementation of the system.

Further, Srinivasan *et al.*, (2004) proposed a methodology for mode and transition identification in chemical processes that forms a fundamental basis of many of the optimality assessment research contributions. Srinivasan *et al.*, (2004) made it clear that process data contains both transition and quasi-steady data, which needs to be analyzed separately. Thus a method of segmenting transient data from quasi-steady data was developed, as described by Srinivasan *et al.* (2004). Transitions were then identified by means of dynamic principal component analysis, the angles between principal components were used as a metric of similarity. For mode identification, Srinivasan *et al.* (2004) made use of the absolute distance of the score means of different windows, which was then compared to a threshold for identification. This technique however fails to take into account the covariance of the data analyzed, as well as the fact that online implementation of the methods is not really discussed.

To account for the covariance of modes Ye *et al.*, (2009) made use of GMMs and the EM algorithm to estimate its parameters. Transient data was first filtered out by means of steady state analysis and then the GMM methodology was applied to only the steady data. Further, the data assigned to the various Gaussians was characterized by normality testing, such as kurtosis and skewness. Safety and optimality indices were then determined for the various modes. The safety index was based on the Mahalanobis distance of the sample to the mode, the smaller the distance, the safer the operation. The optimality index was based on profit and product quality while taking into account the distribution of the mode. This investigation however did not to reduce dimensionality prior to GMM, thus this technique may suffer from the curse of dimensionality. No consideration was also employed on how to determine the number of modes, which is an important issue. Further, the optimality index derived did not take into account mode switching constraints, thus reducing its usefulness. An important issue however raised by Ye *et al.*, (2009) is the post-analysis of the data assigned to the various Gaussians. Thomas, Zhu and Romagnoli, (2018) determined that unsupervised metrics evaluating the performance of cluster results are important

due to the fact that the ground truth of process data is usually not known, kurtosis and skewness are examples of such metrics (Mardia, 1970). Further attempts of optimality assessment with the use of GMMs have been made by Liu *et al.*, (2015). Here both modes and transitions are monitored with the use of GMMs and a comprehensive economic index. Gaussian process regression (GPR) is also implemented to predict the economic index. This methodology has also been implemented on a gold hydrometallurgy process (Liu, Wang and Chang, 2018). No consideration is however placed on dimensionality reduction, and as a result, the techniques may suffer from the curse of dimensionality. Further, data is segmented into modes and transitions manually, therefore limiting industrial application. Although methods of non-optimal cause identification are implemented, mode switching constraints are not taken into consideration.

A different methodology optimality assessment has been implemented by Liu, Wang and Chang, (2016). Here the tedious process of aligning comprehensive economic indices with process data can be avoided by making use of MsPCA (Multiple Set PCA). The idea behind MsPCA is to reveal common basis vectors among multiple datasets and as a result are able to remove common process variations shared over multiple modes. Extracting optimality-related variations, therefore, becomes possible, allowing for easier non-optimal cause identification. The technique was however implemented in a supervised manner, as with the previous techniques, limiting its industrial application. Ying *et al.*, (2020) also made use of MsPCA to characterize the independent characteristics of different datasets for optimality assessment. Subtractive clustering was however also used prior to MsPCA to remove transition data, making this procedure more applicable in an industrial setting. Features obtained from MsPCA are then used to map data onto a 2D grid by making use of self-organizing maps (SOM). SOM is a neural network that can be trained with high dimensional data, the output of which is a 2D map where different regions represent different states as well as the fact that the local neighborhood of the data is more efficiently preserved. This technique is, therefore, ideal for multistate data visualization and evaluation. Similar to previous techniques, an economic index is assigned to the various modes. This technique therefore improves on previous techniques, since a more unsupervised approach is implemented. The determination of the number of modes and mode switching constraints are however crucial issues not discussed. Further, normalized data was used as an input to the SOM, due to the fact that features obtained from MsPCA were used. The scaling parameters of multistate operations are dominated by the large variations caused during transitions and as a result may obscure significant changes within steady states (Ng and Srinivasan, 2008).

It is therefore clear that a lot of the knowledge gained from fault detection and diagnosis literature has been carried over and implemented for process optimality assessment. Similar simplifying assumptions about simulated process data are however also made to demonstrate the feasibility of a technique. Usually, this results in the implementation of unsupervised techniques in a supervised manner, thus limiting the applicability for actual process data.

2.7 Literature Highlights

Existing literature stemming from process monitoring and optimality assessment could be utilised to achieve the stated objectives. Key issues have been identified as process data is high dimensional, multimodal (non-linear), patterns/states within the data are unknown and data may contain process states irrelevant to the analysis. Typically, developed techniques addressing these issues are tested on CSTR or Tennessee Eastman process simulations.

However rarely does an author address all issues within a single article. As a result, to achieve the objectives of this investigation key concepts originating from the various contributions should be integrated. PCA seems like a valid technique to reduce the dimensionality of process data, minimising the variance-bias trade off when estimating the parameters of the various modes (Choi, Park and Lee, 2004). GMMs and the EM algorithm could serve as a useful approach to modelling the multimodality of the process data, such that modes can be discovered in offline implementation as well as assist with state identification in online implementation (Yu, 2011). An optimality measure can be assigned to each mode based on process profit (Liu, Wang and Chang, 2015). Stationarity analysis could be utilised to filter transient states from process data prior to the analyses, thus assisting in the discovery of process modes or steady states(Srinivasan, Wang and Ho, 2004; Kelly and Hedengren, 2013). This could in turn improve the GMM parameter estimates of the various process modes.

Further, the implementation of stationarity analysis could make the various benefits of window-based analyses attainable. Knowledge of mode shifting constraints as well as procedures required to switch modes could be extracted from transient data sequences. The sequence of time series data is a key process data characteristic rarely leveraged in literature.

3 METHODOLOGY

Based on the objectives of this investigation, the state-based decision support system should discover modes from historical data (training data), identify the state (specific mode or transient) a process is in (on testing data), and its economic optimality. A more optimal mode to shift to and the procedures required for the mode shift can then be provided to human supervisors. In order to evaluate the performance of such a state-based decision support system (described in 3.2 later), multimodal simulated data has to be generated, of which the ground truth states are known.

3.1 Case Study for Data Generation: Production of Propylene Glycol in a CSTR

To evaluate the performance of the state based decision support system, synthetic data of which the ground truth is known has to be generated. Various benchmark processes have been implemented to simulate multimodal industrial data, the most popular of which are the Tennessee Eastman process (Zhang *et al.*, 2015; Ying, Li and Yang, 2020) and Continuous Stirred Tank Reactors (CSTR) (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019).

Specifically in this investigation the propylene glycol CSTR described by Fogler (2018) is adapted to simulate multimodal process data. Propylene glycol (C) is produced by the hydrolysis (B) of Propylene Oxide (A), a reaction which occurs readily at room temperature. Since the reaction is a first order (Equation 3-1) exothermic reaction, CSTR cooling is required via cooling water. The process flow diagram of which can be seen in Figure 7.

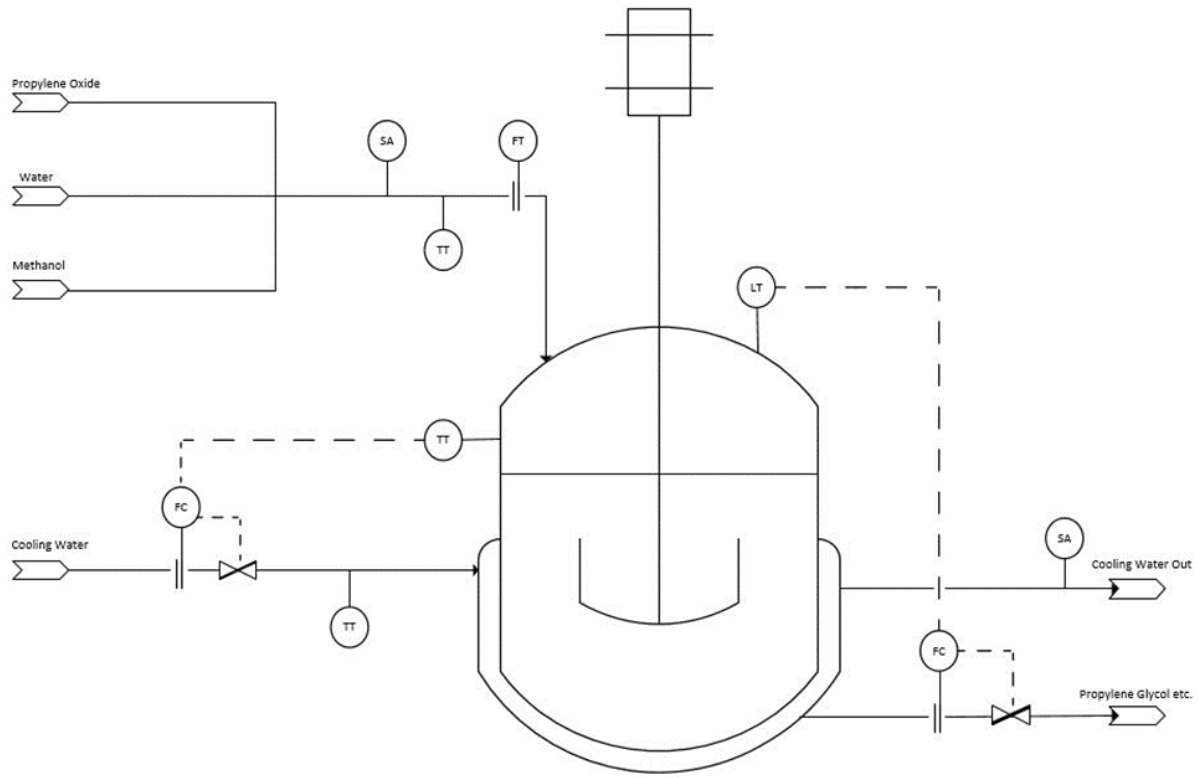


Figure 7: Propylene glycol CSTR process flow diagram

$$r_a = -k_o e^{-\frac{E_A}{RT}} C_a \quad 3\text{-}1$$

Here, r_a describes the reaction rate of propylene oxide with the reaction rate constant k_o, CSTR temperature T, activation energy E_A and propylene oxide concentration C_a. The mole balances of the various reaction components can be seen in Equations 3-2 to 3-5. It should be noted that a solvent, methanol (M), is also fed into the reactor.

$$\frac{dC_a}{dt} = r_a + \frac{V_{in}(C_{ao} - C_a)}{V} \quad 3\text{-}2$$

$$\frac{dC_b}{dt} = r_a + \frac{V_{in}(C_{bo} - C_b)}{V} \quad 3\text{-}3$$

$$\frac{dC_c}{dt} = -r_a - \frac{C_c V_{in}}{V} \quad 3\text{-}4$$

$$\frac{dC_m}{dt} = \frac{V_{in}(C_{mo} - C_m)}{V} \quad 3\text{-}5$$

$$\frac{dV}{dt} = V_{in} - V_{out} \quad 3\text{-}6$$

$$V_{in} = \frac{F_{ao}}{\rho_a} + \frac{F_{bo}}{\rho_b} + \frac{F_{mo}}{\rho_m} \quad 3\text{-}7$$

$$F_{ao} = C_{ao} V_{in} \quad 3\text{-}8$$

Here the CSTR volume, volumetric inlet and outlet flowrates are described by V, V_{in} and V_{out}. The molar densities of propylene oxide, water and methanol are described by ρ_a, ρ_b, ρ_m. The CSTR is also non-isothermal, Equations 3-9 to 3-12 describe the energy balance within the CSTR. Here, ΔH_{Rx} describe the heat of reaction and vaporisation. T_o and T_c refer to the inlet temperatures of the reactants and cooling water. All Cp terms refer to the heat capacities of CSTR constituents. Further, m_c refers to the molar flowrate of the cooling water and UA refers to the heat transfer coefficient of the heat exchanger.

$$\frac{dT}{dt} = \frac{Q_g - Q_r}{C_a V C p_a + C_b V C p_b + C_c V C p_c + C_m V C p_m} \quad 3\text{-}9$$

$$Q_g = r_a V \Delta H_{Rx} \quad 3\text{-}10$$

$$Q_r = F_{ao}\theta(T - T_o) + m_c C p_b (T - (T - T_c) e^{-\frac{UA}{m_c C p_b}} - T_c) \quad 3\text{-}11$$

$$\theta = Cp_a + \frac{F_{bo}}{F_{ao}} Cp_b + \frac{F_{co}}{F_{ao}} Cp_c + \frac{F_{mo}}{F_{ao}} Cp_m \quad 3\text{-}12$$

Further, it is assumed that the economic performance of the CSTR is only a function of the molar flowrates of propylene glycol, propylene oxide and the cooling water. Equation 3-13 describes the economic

performance (P_{CSTR}), where p_{glyc}, $p_{prop\ oxide}$ and $p_{cooling}$ are cost parameters described in Appendix D Table 28 and M denotes the mass number of the considered components.

$$P_{CSTR} = C_c V_{out} M_{glyc} p_{glyc} - C_a V_{in} M_{prop\ oxide} p_{prop\ oxide} - \frac{m_c}{\rho_b} p_{cooling} \quad 3\text{-}13$$

Additional information about the CSTR initial conditions and parameter values can be found in Appendix D Table 28 and Table 29. It should be noted that all simulations contain a reactor start-up period as discussed by Fogler (2018), thus more accurately reflecting industrial data. Data samples from start-up or shut-down periods may be very different to "normal" operation. A dataset containing many such samples ("outliers") may challenge the analysis approach.

Further adaptations to the described model by Fogler (2018) include temperature as well as level control. CSTR temperature control is achieved by manipulating the cooling water flowrate. Level control is achieved by manipulating the CSTR outlet flowrate. Both controllers are proportional integral feedback controllers, parameters of which can be seen in Appendix D Table 30. The disturbance variables within this simulation were chosen to be the inlet flowrates of the reactants, the CSTR inlet temperature (T_o) and the cooling water temperature (T_c). The described model can be found at https://github.com-/FrancoisNoelle/DecisionSupport.

3.1.1 Multimodal Data

Multimodal training and testing data is generated randomly within this investigation. As discussed each CSTR simulation contains an initial start-up period, after which random modes of random duration occur. These random transitions to modes may result either due to changes in the controller set points or sustained changes within the disturbance variables. It should be noted that the transitions are driven only by a single change, for example the set point of the level controller may step, but the set point of the temperature controller has to remain constant. Similarly, only a single sustained disturbance may occur for each transient. Multimodal industrial process data will, however, not have the same characteristics. Various set point changes may occur continually during process operation due to more complex hierarchical control systems etc. This assumption was formulated such that the "causal" analysis procedures required (determination of cause of the transition) would not have to be complex. Sophisticated causality analysis is outside the scope of this investigation.

Set point changes for the temperature controller are implemented via a step change. Set point changes for the level controller are however implemented via a first order filter, such that abrupt changes within the CSTR do not occur (Miskin, 2016). Similarly all changes in the disturbances are implemented via first order filters.

$$\tau \frac{dI}{dt} - I = f(t) \quad 3\text{-}14$$

Equation 3-14 describes how certain CSTR input variables transition, where I describes the input variable, τ describes the time constant of the input variable transition and $f(t)$ is the forcing function (step

change) of the various input variables. More information on the various time constants can be found in the Appendix D Table 32.

Multimodal training data can thus be generated given a simulation duration, minimum steady state period, randomising seed (for repeatability) and high/low values for all input variables (set points and disturbance variables). A visual representation of the data generation procedure can be seen in Figure 8. The minimum steady state period is required to ensure that the CSTR reaches a quasi-steady state or mode after a transition is induced via Equation 3-14. The high/low pairs for all input variables provide reasonable options for the various input variables, such that the CSTR remains stable as well as all unique input variable pairings can be considered as different modes. The randomising seed allows for simulation results to be repeatable. A more comprehensive look as to how the random multimodal data is generated can be seen in https://github.com/FrancoisNoelle/DecisionSupport.

The multimodal testing data is generated in the same multimodal sequence as the training data, ie. the same modes (input variable pairings) occur sequentially in time as with the training data. This was done so that no new modes occur within the testing data. Adaptive extensions of the decision support system are outside the scope of this investigation. The duration of the modes however varies randomly and thus the testing data serves as an effective dataset to evaluate the state based system.

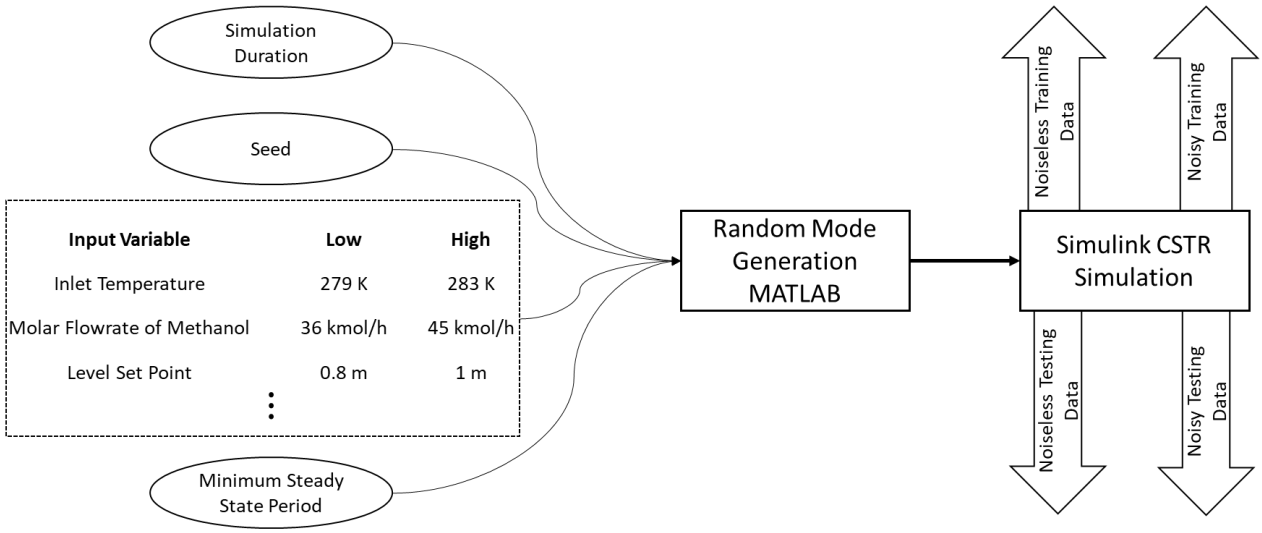


Figure 8: Schematic describing how multimodal random CSTR training and testing data is generated

3.1.2 Ground Truth Determination

As seen in Figure 8, four sets of data are generated for each simulation run. For both the training and testing data, noisy and noiseless CSTR data is generated. Here noisy data refers to CSTR data which contains both process noise and measurement noise. Noiseless/noise free data contains neither and is purely generated for the purpose of automating the ground truth determination, such that the state based decision support system can effectively be evaluated.

All input variables for the noisy CSTR simulations (testing and training) are simulated to have first-order autoregressive process variations in them (Equation 3-15). Additionally, white noise on the output/measured variables is also included within the simulation (Equation 3-16). As a result the serially

correlated properties often seen in industrial process data are more effectively simulated. These process variations therefore make up the "common-cause" variations of the CSTR (Yoon and Macgregor, 2001).

$$x_t = \phi x_{t-1} + \sigma_e \quad \textit{3-15}$$

$$x_{t,meas} = x_t + \sigma_m \quad \textit{3-16}$$

The auto regressive coefficients (ϕ) for the various input variables, the process noise (σ_e) and the measurement noise (σ_m) can be found in Appendix D Table 31. Inclusion of these common cause variations thus simulate properties of real processes more effectively, but also make the determination of the ground truth more difficult. In the case of this investigation the ground truth of the CSTR data refers to the true state the CSTR is experiencing, ie. transient state or the specific mode.

The ground truth of a process is defined by the input variable pairings (high/low), such that each unique input variable pairing is a different mode. Determining the mode of the CSTR is thus relatively simple, however since the CSTR experiences relatively complex dynamics, determining the ending times of transitions is difficult. This is where the noiseless simulations are useful. The procedure for determining the ground truth state of the CSTR training data is as follows.

1. Determine the mode pairings, each unique input variable paring is a different mode

2. Assign each sample in time to its mode, not yet considering the transitions between states. Modes are for now indexed based on their sequence in time.

$$s_t = unique\ mode\ index \quad \textit{3-17}$$

3. Use the noiseless training simulation to determine the duration of the transient periods. The CSTR is considered to be steady once all input variables (disturbance variables/controller set points) are close to their set values (within 99 %) for an extended period of time.

$$if\ \left|x_{t,noiseless} - x_{t,set}\right| \leq \left|0.99 x_{t,set}\right| \text{ then } Y_{t,ground\ truth} = 1 \quad \textit{3-18}$$
$$else$$
$$Y_{t,ground\ truth} = 0$$

4. Determine the product of $y_{t,ground\ truth}$ and s_t for all samples, thus the transient periods are included within the ground truth.

$$s_t = Y_{t,ground\ truth} \times s_t \quad \textit{3-19}$$

5. Calculate the fraction each mode makes up of the training data. Redefine the mode identifying indices based on their fraction of the data in descending order. In other words the "heaviest" or most frequently occurring mode is assigned index one etc. The transient state identifying index remains zero.

The ground truth of the testing data is determined in exactly the same manner, however the unique mode identifying indices obtained from the training data are utilised. Thus, the ground truth of the CSTR simulations can be implemented to evaluate the effectiveness of the state based system. The algorithmic

form of the ground truth determination procedure can be seen in https://github.com-/FrancoisNoelle/DecisionSupport.

3.2 Algorithms

The overall proposed state-based decision support system makes use of various techniques to provide actionable advisories. PCA, stationarity analysis, K-means, GMMs, sequential analysis and finally key performance indicators are utilised. The overall analysis procedure can be sub-divided into five sections: PCA, stationarity analysis, state analysis, connectivity analysis and online implementation.

3.2.1 Principal Component Analysis Application

PCA serves as a dimensionality reduction technique within this investigation, thus avoiding issues discussed in 2.3.2.1. The procedure of determining the principal components from the training data is described in Table 1, the code of which can be found https://github.com-/FrancoisNoelle/DecisionSupport.

Table 1: PCA Procedure

Input: Training data matrix in format described by Equation 2-1 and variance to be retained (hyper parameter)
Output: Retained PCs, PC scores and normalization parameters
1. Normalize training data using Equations 2-2, 2-3 and 2-4
2. Compute the covariance matrix using Equation 2-9
3. Solve the eigenvectors (PCs) and eigenvalues (variance explained) of the covariance matrix via eigenvector decomposition (Equation 2-10)
4. Using Equations 2-13 and 2-14 determine the number of PCs to retain
5. Project training data into the retained PC space ie. determine PC scores using Equation 2-15

3.2.2 Stationarity Analysis

Stationarity analysis is only implemented on training/offline data within this investigation. SSD serves the purpose of removing transients prior to the state analysis, making the overall process more robust. The procedure described in Table 2 is performed for each separate window, as displayed in Figure 9.

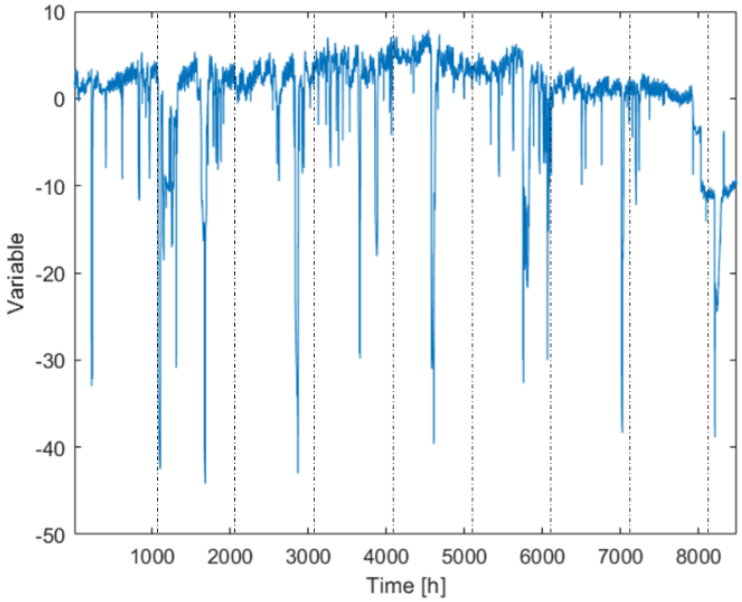


Figure 9: Diagram showing the various windows (segments separated by dashed lines) SSD is performed on (window size is exaggerated to improve interpretability)

Table 2: Window Based SSD

Input: Training Data (Original Variables/Scores), Window Size (n), Significance (α), Steady State Threshold (θ_{SS})
Output: Binary indication of steadiness for each sample, probability of each window being stationary
1. Estimate the data window drift component using Equation 2-36
2. Estimate the window mean component using Equation 2-37
3. Estimate the window variance using Equation 2-38
4. Perform the students t-test as described by Equation 2-39 for each sample within the window
5. Repeat steps 1 to 4 for all considered variables
6. Extend this SSD to the multivariate case by implementing 2-40
7. Deem the window steady or transient using Equation 2-41

3.2.3 State Analysis

Here the various modes within the training data are discovered, making use of mainly unsupervised clustering techniques. It should be noted that a novel GMM refining procedure is introduced in Table 3, which is based on the sequence of the time series data. Here, the frequency of sequential mode switching

is utilised as a metric, indicating if modes within a GMM should be merged. The idea is that processes are usually operated within a specific mode or quasi-steady state, excessive switching between various modes could indicate that a single Gaussian should rather describe various Gaussians within the GMM.

Table 3: Procedure for determining the State of Time Series Data

Input: Training Data (Normalized Original Variables/Scores), SSD results (optional), Percentile
Output: Clustered Training Data (m_t), Gaussian Mixture Model ($\boldsymbol{\Theta}$), Local NLLP Thresholds ($\boldsymbol{NLLP}_\alpha$)
1. Optional: Remove all transient periods from training dataset ($\mathbf{X}_{Adj}$)
2. Determine the number of stationary periods within the SSD analysis (alternatively set a maximum number of clusters to consider). This value is C_M
3. Determine the C_M initialisation parameters for the various GMM configurations
i) Preform K-means
ii) Use the cluster centres, covariances and weights as initialisation parameters θ_0 (Equation *2-25*). Alternatively, initialisation parameters can be setup manually with expert knowledge along with SSD results.
iii) Fit a GMM to the training by performing E-step (Equation 2-26) and M-step (Equations 2-27 to 2-29) of the EM algorithm.
iv) Perform the EM algorithm until Equations 2-30 and 2-31 are satisfied (below tolerance)
v) Calculate the BIC of the GMM configuration using Equation 2-34.
4. Select the GMM configuration from the C_M options with the lowest BIC
5. Rearrange the mode index i in $p(x\|\theta)$ (Equation 2-23) based on the descending magnitude of w_i
6. Refine the chosen GMM based on the sequence of data
i) Determine the mode of each training sample using Equation 2-26, the Gaussian with the maximum posterior probability defines the mode of the sample (m_t)
ii) $For\ all\ n-1\ samples$ if $m_t \neq m_{t+1}\ then\ ew_{m_t m_{t+1}} = ew_{m_t m_{t+1}} + 1$ Here ew_{ij} (edge weight) defines the number of sequential transitions from mode i to mode j A connectivity matrix ($\mathbf{EW}^{K\times K}$) can then be setup in the form as described by Equation 3-20, any switching conditions that do not occur are set to zero (along with diagonals) $$\mathbf{EW} = \begin{bmatrix} 0 & ew_{12} & \dots & ew_{1K} \\ ew_{21} & 0 & \dots & ew_{2K} \\ \vdots & \vdots & \dots & \vdots \\ ew_{K1} & ew_{K2} & \dots & 0 \end{bmatrix} \quad \textit{3-20}$$
iii) $If\ ew_{ij} \geq sequence\ threshold\ then\ modes\ i\ and\ j\ are\ merged$, alternatively ew_{ij} can be used as guidance for manually selecting modes to merge.
iv) θ_0 is updated with the new information

v) Perform the EM algorithm until Equations 2-30 and 2-31 are satisfied (below tolerance)
vi) Rearrange the mode index i in $p(x|\theta)$ (Equation 2-23) based on the descending magnitude of w_i

7. Refine the chosen GMM based on the Gaussian weights (w_i)

i) Determine the mode of each training sample using Equation 2-26, the Gaussian with the maximum posterior probability defines the mode of the sample (m_t)
ii) If $w_i \leq weight\ threshold\ then\ remove\ w_i g(x|\theta_i)\ from\ GMM$
iii) Further remove all m_t which equal i, thus removing these samples from training data ($\mathbf{X}_{Adj}$)
iv) θ_0 is updated with the new information
v) Perform the EM algorithm until Equations 2-30 and 2-31 are satisfied (below tolerance) or simply update remaining w_i
vi) Rearrange the mode index i in $p(x|\theta)$ (Equation 2-23) based on the descending magnitude of w_i

8. Determine the best matching cluster/Gaussian (BMC) for all samples in $\mathbf{X}_{Adj}$ as in step 6 i)

9. Calculate the NLLP of the various samples $(\mathbf{X}_{Adj})$ using Equation 2-32. Use the input percentile of the NLLP assigned to a mode (BMC) as the local mode threshold (Addo, 2019), denoted $NLLP_{i\propto}$

10. Determine the mode of all samples in the training dataset ($\mathbf{X}$)

i) Determine a samples BMC as in 9. i), $i = BMC$
ii) Determine the $NLLP_t$ of the sample using Equation 2-32
iii) $if\ NLLP_t \leq NLLP_{i\propto}\ then\ m_t = i\ else\ m_t = 0$

Table 3 provides an overview of the entire state based procedure, it should however be noted that certain steps within the procedure can be omitted or switched. For example GMM refinement procedures could in fact be skipped or Gaussian deletion could occur prior to merging. The specific procedure/thresholds implemented depend on the knowledge the expert has of the process.

3.2.4 Connectivity Analysis

Within the context of this investigation connectivity analysis entails assigning a key performance indicator to the various modes and determining the conditions required to switch from one mode to another.

Table 4: Procedure for Mapping Process States

Input: SSD results, mode indices (m_t) , Training Data ($\mathbf{X}$) and the KPI function

Output: Final Actionable Advisory Model

1. Decide which variables determine the performance of the process. Within the context of the CSTR it is decided that the performance function holds the form as seen in Equation 3-13

2. Determine the KPI for each unique mode ($KPI_{avg\ for\ mode\ i}$) using Equation *3-21* (Marlin, 1995)

$$KPI_{avg\ for\ mode\ i} = \sum_{z=1}^{I} P_z F_z \qquad 3\text{-}21$$

Here I denotes the number of intervals which the KPI is divided into. P_z denotes the KPI or performance at the midpoint of an interval. F_z denotes the fraction of data (of mode i) within the interval z

3. Using the SSD results combine all succeeding stationary windows (no transient windows in between) into a single stationary period ($t \rightarrow t + s$), here s denotes the duration of a stationary period

4. Assign a single mode to all stationary periods ($s_{t \rightarrow t+s}$) using Equation 3-22 ($mode$ here refers to the statistical mode ie. state that occurs most often within period)

$$sm_{t \rightarrow t+s} = mode(m_{t \rightarrow t+s}) \qquad 3\text{-}22$$

5. As with Equation 3-20 setup a connectivity matrix. Here however not only the connectivity but also the transition "driving force" between the various stationary periods is setup.

$$if\ sm_{t_1 \rightarrow t_1+s_1} \neq sm_{t_2 \rightarrow t_2+s_2}\ then\ \mathbf{MAP}_{(sm_{t_1 \rightarrow t_1+s_1})(sm_{t_2 \rightarrow t_2+s_2})} = 1 \qquad 3\text{-}23$$

6. Determine if the transition during $t_1 + s_1$ and t_2 occurred as a result of a disturbance or a set point change. If a set point change occurred during the transient, then it is assumed that the driving force of the transient is that specific set point change, else it is disturbance related. This information is logged within **MAP**. This process can be seen in more depth at https://github.com/FrancoisNoelle/DecisionSupport.

7. Both the switching conditions as well as the KPIs of the modes are now known. A connectivity diagram containing the switching constraints (disturbance caused and controller caused transients are distinguished) and the mode KPIs can now be drawn.

3.2.5 Online Implementation and Providing Actionable Advisories

Table 5: Procedure for Online implementation of Trained State Based Model

Input: Testing/Real Time Data
Output: Actionable Advisories

1. Normalize testing data using output normalization parameters from procedure 3.2.1 and Equation *2-4*

2. If PCA was performed during training, project normalized testing data into PC space using Equation 2-15 and the retained PCs output from procedure 3.2.1

3. Using the GMM output from procedure 3.2.3, determine the BMC for the current sample in time

4. Determine the NLLP of the sample and if it lies below the threshold $NLLP_{i\propto}$

$$if\ NLLP_t \leq NLLP_{i\propto}\ then\ m_t = i\ else\ m_t = 0$$

5. Using the model/map output in 3.2.4, determine the optimality of the current mode using Equation 3-24 (adapted from Liu *et al.*, 2018)

$$Process\ Optimality = \frac{KPI_{avg\ for\ mode\ m_t}}{\max accessible\ mode\ KPI} \qquad 3\text{-}24$$

Here $\max accessible\ mode\ KPI$ refers to the maximum KPI of the modes accessible via a set point change not a disturbance, thus taking into account the mode switching constraints. No advisories are provided if the process is within a transition or an unknown state.

6. Actionable advisories therefore entail the current state the process is experiencing, the optimality of the state, which state would be more economically favourable and how to reach this more favourable state

---

The overall data analysis procedure is described by Figure 10. Here the flow of inputs and outputs resulting from procedures described in Table 1, Table 2, Table 3, Table 4 and Table 5 are shown, thus showing the interconnectivity of the various processes.

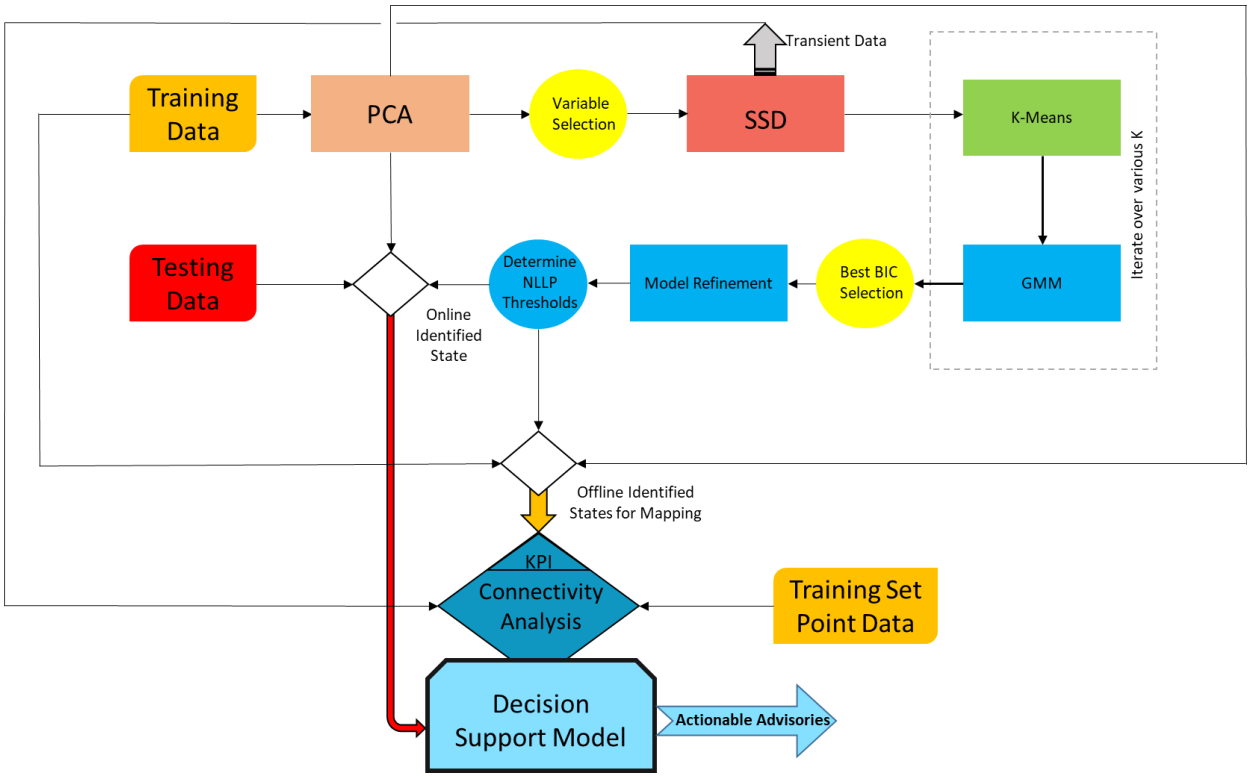


Figure 10: Overall state-based decision support system approach where all procedures discussed in 3.2 are integrated

3.3 Evaluation of the Proposed State Based Approach on CSTR Simulation Data

A fundamental aspect of the evaluation of any monitoring/state based model is selection of suitable performance metrics. Within the context of this investigation, two critical procedures have to be evaluated, steady state analysis and the state based analysis.

3.3.1 Steady State Detection Performance Evaluation

Since the output of a SSD algorithm is binary, a sample can be either steady or transient, therefore, evaluation techniques can be adapted from fault detection approaches. Utilising the ground truth stationarity obtained as described in 3.1.2, the missed and false transient rates can be determined via a confusion matrix. This confusion matrix can be seen in Table 6. Since SSD is only preformed on training data, its effectiveness is only evaluated on training data.

Table 6: SSD Confusion Matrix

		Detected State	
		Steady	**Transient**
True State	**Steady**	TP_1	FP
	Transient	FN	TP_2

The false transient rate (FTR) which can be related to the false alarm rate in fault detection is therefore calculated using Equation 3-25.

$$FTR = \frac{FP}{TP_1 + FP} \qquad 3\text{-}25$$

Here FP and TP_1 denote the false positives (stationary based on ground truth, detected as transient) and true positives (correctly identified as steady). Similarly the missed transient rate (MTR) can be determined using Equation 3-26.

$$MTR = \frac{FN}{TP_2 + FN} \qquad 3\text{-}26$$

3.3.2 State Identification Evaluation

Like fault diagnosis, state identification is not a binary analysis procedure, since various classes may exist. Thus the use of a multi class confusion matrix is required to evaluate the performance of the mode identification procedure. This multiclass confusion matrix (Table 7) is determined from the ground truth of the testing data determined as discussed in 3.1.2. Thus simulates the industrial procedure, model obtained from historic data and utilised on real time/testing data.

Table 7: Example Multiclass confusion matrix consisting of i classes

		Estimated State Label			
True State Label		1	2	••	i
	1	TP_1	E_{12}		E_{1i}
	2	E_{21}	TP_2		E_{2i}
	⋮				
	i	E_{i1}	E_{i2}		TP_i

As can be seen in Table 7 the diagonals of the confusion matrix describe correct state identifications. The off diagonal entries in Table 7 describe the number of times a state was falsely identified. For this investigation it was decided to implement multiclass performance metrics precision and recall. Both these measures are quite popular within the context of multimodal process monitoring (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019).

Precision refers to the fraction of correct state (specific mode/transient) identifications made relative to the number of times the specific mode was retrieved, thus a precision metric is determined for each state via Equation 3-27.

$$Precision_i = \frac{TP_i}{sum\ of\ confusion\ matrix\ column\ i} \tag{3-27}$$

Recall on the other hand refers to the fraction of correct state identifications that could have been made within the analysis. A separate recall metric is obtained for each state using Equation 3-28.

$$Recall_i = \frac{TP}{sum\ of\ confusion\ matrix\ row\ i} \tag{3-28}$$

To avoid the analyses of each performance metric separately, recall and precision are combined into a single metric known as the F_1 score. The F_1 score or the F-measure is the harmonic mean between precision and recall and is thus determined using Equation *3-29* (Sasaki, 2007), where recall and precision are weighted of equal importance.

$$F_{1_i} = \frac{2(Precision_i Recall_i)}{Precision_i + Recall_i} \tag{3-29}$$

To determine a single F_1 score for the entire state identification system, the macro average of the various F_1 scores is determined using Equation 3-30.

$$F_{1_{macro}} = \frac{\sum_{i=1}^{K+1} F_{1_i}}{K+1} \tag{3-30}$$

Here once again, K denotes the total number of modes. However since there are $K+1$ states due to the presence of transition data, the additional state has to be included. The code for this evaluation procedure can be found https://github.com/FrancoisNoelle/DecisionSupport.

4 DEVELOPMENT AND EVALUATION OF THE DECISION SUPPORT SYSTEM ON THE CSTR

Within this chapter some the CSTR simulation data characteristics are described. The decision support system described in Figure 10 is applied and evaluated to two multimodal CSTR simulation datasets, one containing 6 modes and the other containing 15. The effect transient data has on the performance of the state identifying procedure is investigated and compared to if steady state detection was performed prior to the analysis. Further, alternative approaches to steady state analysis are also discussed within this chapter.

4.1 Evaluation on CSTR Simulation Data

4.1.1 Simulation Data Description

A multimodal dataset containing both transient and stationary periods was generated by means of the simulation as discussed in the 3.1.1. A total of 14 variables were measured, their time evolution can be seen in Figure 11. It should be noted that CSTR start up is contained within the dataset, but was omitted in Figure 11 to improve the interpretability of the data.

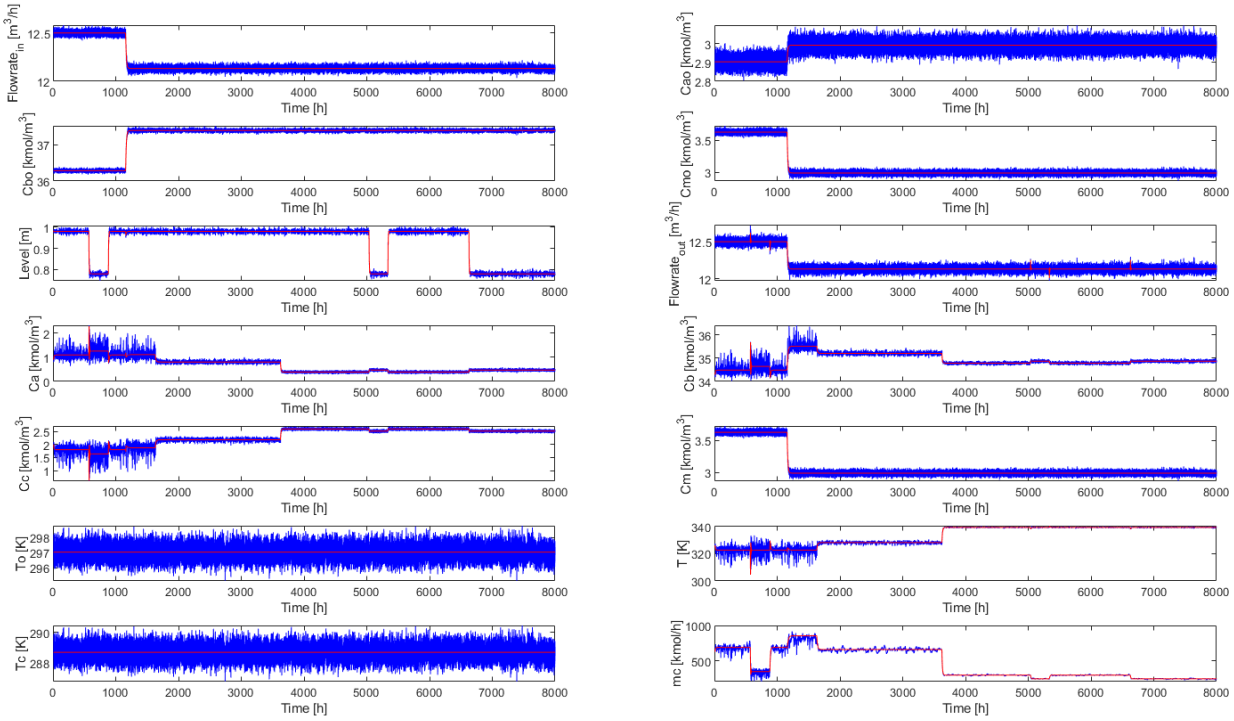


Figure 11: Multimodal Data generated using CSTR simulation over a duration of 8000 hours with a seed 60, where the red line indicates the simulation results without noise and the blue describe the simulation results with noise

This dataset contains 6 unique modes and various transitions between these modes as described in 1.2. Each mode can be identified from the input variables, which in this case were the CSTR temperature and level controller set points, inlet propylene oxide flowrate, inlet reactant water flowrate and the inlet methanol flowrate. The reactant feed and cooling water temperatures were also mode defining input variables. Disturbance variables and set points are therefore the mode defining factors. Unique pairings

of these variables define the various modes. The dataset displayed in Figure 11 resulted from the input variable pairings described in Table 19 in Appendix A. It should be noted that Table 19 may repeat certain pairings, in these cases no transitions occurred. Figure 11 shows both the noisy and noise free simulation results, indicated by the blue and red lines. It is therefore clear that the auto correlated process noise and measurement noise have a considerable impact on the results.

Careful analysis of all variables in Figure 11 allows for the six modes to be identified as described in 1.2. However, due to the large number of variables that need to be considered, this is not an easy task, in the offline case and especially if the modes need to be identified online. Literature discussed in 2.6. is therefore justified, resulting in less effort required to analyse multivariate/multimodal process data.

Further, it can be seen in Figure 11 that certain variables reflect the multimodality of the dataset more effectively than other variables. The cooling water flowrate (more clearly seen in Figure 12), which is the manipulated variable for the temperature control of the CSTR, seems to be one of the more indicative variables displaying the various modes.

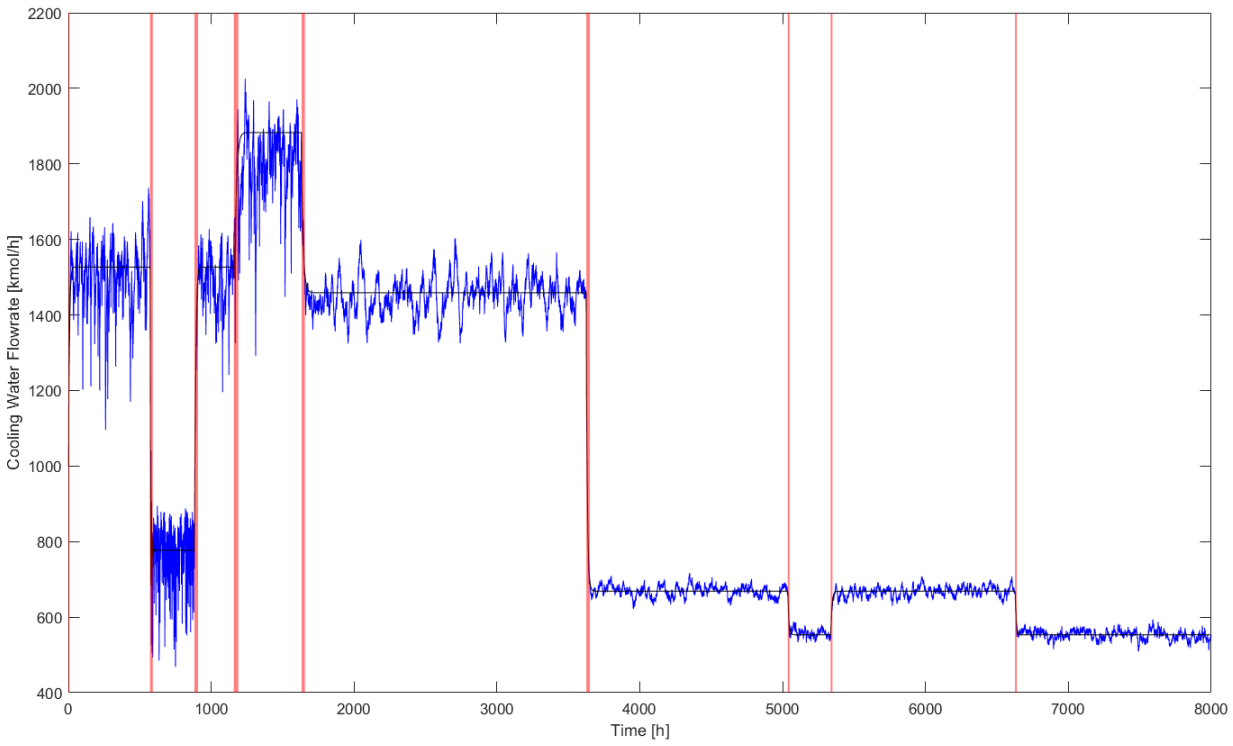


Figure 12: Closer look at the cooling water flowrate displaying the 6 modes, where the blue and black lines indicate the noisy and noiseless simulation results. The red vertical lines are actually shading indicating the transient periods.

The red lines in Figure 12 are actually shaded regions displaying the transient states. Due to the fact that the transient periods make up such a small fraction of the entire dataset, they appear as lines. Figure 13 clearly shows some of the transitions occurring within the process data. These transient periods were

determined from the noise free simulation as described in the 3.1.2. In Figure 13 a) and b) a magnified representation of some of the transients occurring in the simulation can be seen.

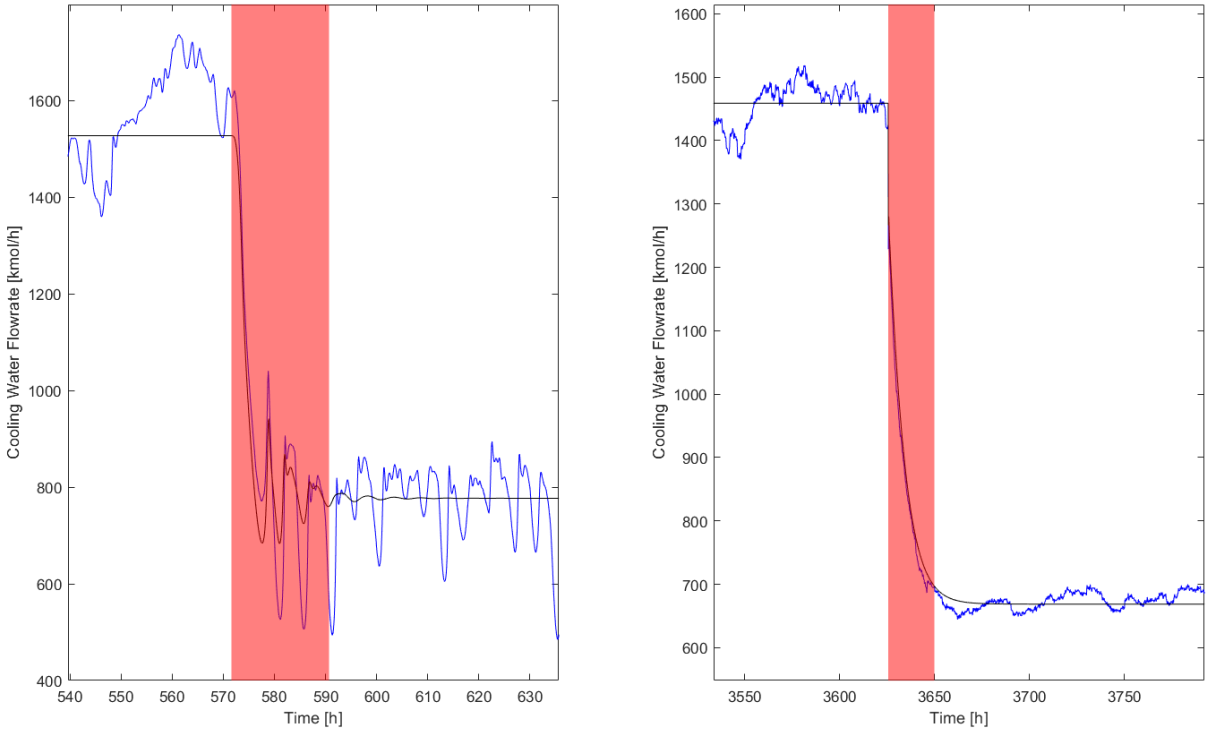


Figure 13: a) Magnified representation of transition occurring 570 hours b) Magnified representation of transition occurring 3620 hours, where the shaded regions represent the true transient periods

Comparing Figure 13 a) and b) it is clear that the transitions occurring here differ not only in duration, but also in the dynamic process response. The approximately 20 hour transient period displayed in Figure 13 a) resulted due to a set point change of the level controller, as described in Table 19 in Appendix A. Figure 13 b) on the other hand displays a 25 hour transient period resulting from a temperature controller set point change. In both cases the temperature controller had to make adjustments to the flowrate of cooling water. A lower CSTR level resulted in a smaller reactor volume and consequently less heat being generated. Less cooling water is therefore required to maintain the CSTR at its set point temperature. Similarly when the set point of the temperature controller is increased, less cooling water is required to reach the desired CSTR temperature. Figure 13 therefore clearly shows that varying dynamic responses occur within the simulated CSTR process data, simulating industrial process data.

Figure 13 also shows the extent of the introduced measurement and process noise on the cooling water flowrate. The difference in responses of the noise free and noisy CSTR simulations is clear. What should also be visible in Figure 11, Figure 12 and Figure 13 is that the common cause variation (mode covariance) occurring throughout the sequence of the process data is not constant, even though the magnitude of variance introduced remains constant. The controllers were tuned to effectively diminish variation in certain modes, however as the process switches modes, the process dynamics also differ therefore the controllers may become more or less effective.

The inertial effect of the autocorrelated process noise introduced in the process variables is also visible in Figure 13, therefore simulating effects seen in actual process data. A graphical representation of the extent of autocorrelation within the process data can be seen in Figure 14.

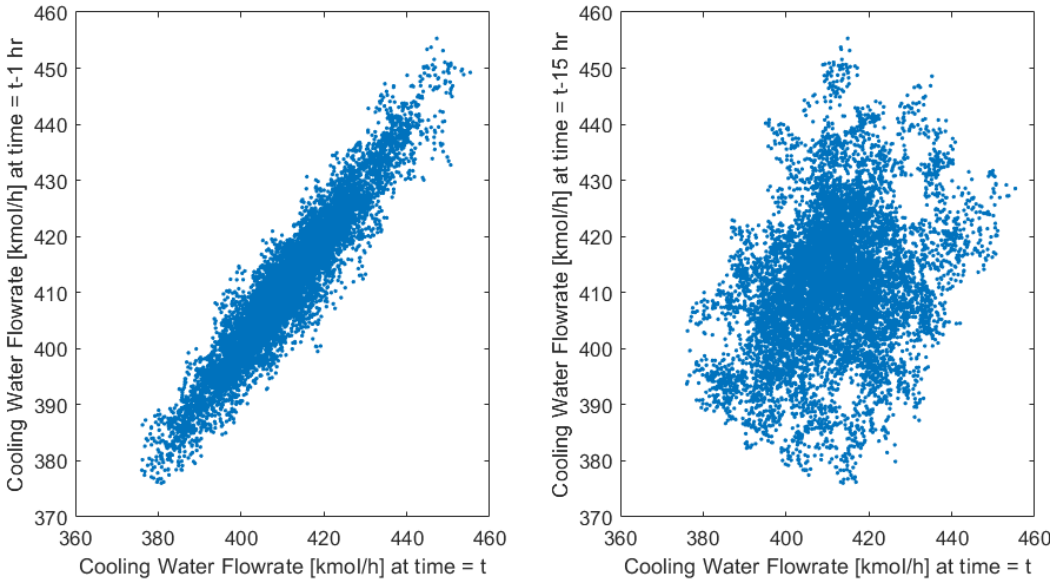


Figure 14: a) Cooling water flowrate plotted against a 1 hour lagged version of itself
b) Cooling water flowrate plotted against a 15 hour lagged version of itself

Figure 14 a) clearly shows that a correlation between the cooling water flowrate and its previous values exists within a single mode or quasi steady state. The persistence or inertia of measurements seen in Figure 13 is therefore more clearly visible in Figure 14 a). As mentioned in 2.3.2.3, autocorrelation poses difficulties in MSPC, as well as in steady state detection (Brown and Rhinehart, 2000). Approaches used to diminish the effects of auto correlated noise on analysis have been to make use of DPCA (not considered within this investigation) or simply increasing sampling interval (Rhinehart, 2013). Figure 14 b) shows the impact of increasing the sample interval, the degree of correlation between subsequent samples clearly decreases.

The simulated dataset considered within this analysis is therefore somewhat representative of datasets that could be found in industry due to its multimodality, complex nonlinear interactions between the variables, controller interactions and varying process dynamics occurring during transitions. Table 8 describes various basic key properties. These properties need to be taken into consideration when applying stationarity analysis to a dataset.

Table 8: CSTR simulation dataset properties, where the simulation seed was 60 and simulation duration was 8000 hours

Longest Stationary Period [h]	Shortest Stationary Period [h]	Longest Transient Duration [h]	Shortest Transient Duration [h]	Sample Rate [samples/h]	Number of Modes	Fraction of Data at Steady State
1970.2	250.3	31.9	7.3	10	6	0.98

4.1.2 Stationarity Analysis

4.1.2.1 Default SSD hyper-parameter settings

Table 9 describes the tuning parameters set as described by the guidelines of Kelly and Hedengren (2013). The guidelines state that key variables considered within the analysis should be the manipulated and controlled variables of the system. For the CSTR the reactor temperature, level, cooling water flowrate and reactor outlet flowrate are thus considered within the multivariate analysis. Recommendations state that the window size should be set to three to five times the time constant of the process. The window considered should not be too short, otherwise the process may not reach stability. The window should however not be too long, otherwise periods of unsteadiness may be deemed to be quasi steady. Process knowledge should be used in setting the significance and threshold parameters, such that the overall tuning works effectively for the application.

Table 9: Default Tuning Parameters for SSD

Window Size (*n*)	Variables	Threshold (θ_{ss})	Significance (α)
7.3×10×5 = 365	All CVs and MVs	0.2	0.05

According to the guidelines it is clear that considerable process knowledge is required for effective tuning. Table 8 describes some of this required knowledge. An issue however is that the process such as in this case has relatively complex dynamics, and as a result cannot be described by a single time constant. Figure 15 visually displays the results obtained from the SSD at its default tuning performed as described in 3.2.2. The performance of the SSD technique is evaluated as described in 3.3.1.

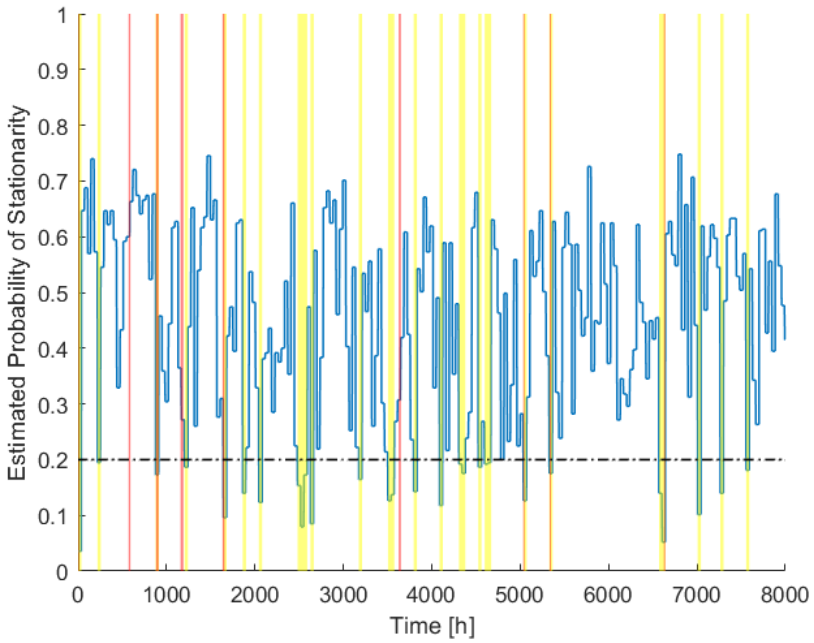


Figure 15: SSD results obtained from default tuning, where yellow and red shaded regions denote detected and actual transient periods. The blue data points describe the probability of a time window being stationary and the dotted line denotes θ_{ss}

At the defaulting tuning the SSD resulted in a MTR (missed transient rate) of 51 % and a FTR (false transient rate) of 12 %. Figure 15 visually agrees with the MTR metric, where many of the actual transients are missed. The transient period detected is however far longer than the actual transient as seen in Figure 16. The estimated probability of a time window being stationary also remains high, even if the given window is actually transient, resulting in the high MTR. This can be seen at 570 hours in Figure 15, where level set point change occurred. The default SSD tuning seems to be less effective in detecting transients resulting from level set point changes.

Various false transients are also detected, such as at approximately 2000 hours seen Figure 15. Here SSD setting seems to be too sensitive, such that measurement and process noise is deemed as transient behaviour. The FTR metric however does not reflect these false detections well, due to the fact that the stationary periods outweigh the transient periods drastically.

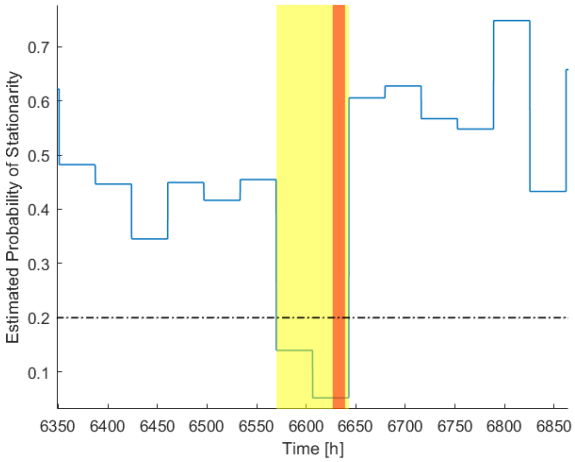


Figure 16: Magnified SSD results displaying that the detected transient (yellow) is far longer than the actual transient (red)

All in all, the default tuning is not very effective as seen in Figure 15, Figure 16 and the high MTR. It is clear that the complex dynamics occurring within the CSTR pose a difficulty to SSD, firstly due to the fact that various types of transitions occur and secondly process noise during quasi steady operation is often indistinguishable from a transition. It is critical that the SSD tuning/technique is able to effectively deal with such issues. Literature however rarely discusses procedures in setting SSD tuning parameters and how sensitive the SSD method is to these parameters (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019).

4.1.2.2 SSD Hyper parameter Investigation

Xu et al. (2018) stated that SSD tuning is crucial for an effective implementation to a specific application. For both online and offline application of various SSD techniques, it is clear that the window size (n) is a key tuneable parameter that needs to be set effectively for accurate detection. Further, the variables considered and the significance are also important parameters that can be tuned such that transients become distinguishable from quasi steady states. Finally, the threshold (Θ_{SS}) is used as the distinguishing parameter. Therefore, if the previous three parameters are not set effectively, setting of Θ_{SS} will also be difficult. The investigation will therefore be structured as follows, first the effect of varying window size will be investigated. Then the effect of varying significance and considered variables at set window sizes will be investigated.

Figure 17 displays the mean estimated probability of stationarity of the actual stationary and transient periods. The analysis was performed on the manipulated and controlled variables with a significance of 5 %. As can be seen in Figure 17, the window size varies from 10 to 2000 samples in size. The threshold (Θ_{ss}) has not yet been included in the analysis.

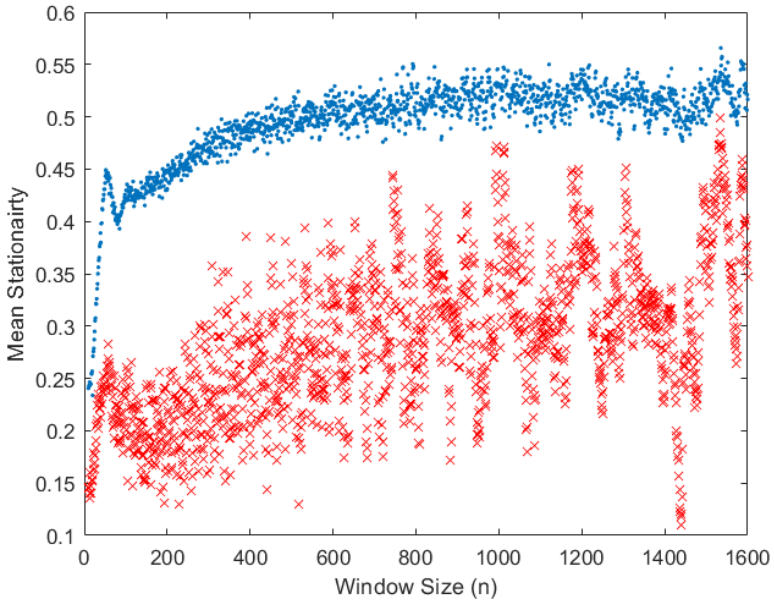


Figure 17: The effect of window size on the mean (averaged over the periods) estimated probability of stationarity of true stationary (dots) and true transient periods (crosses)

Firstly, Figure 17 shows that as the window size changes, the threshold (Θ_{SS}) would also have to be varied such that the SSD remains effective. For example, according to Figure 17 if the SSD was applied with a window size of 600 samples, θ_{SS} would have to be between 0.4 and 0.5. However, if the window contained 200 samples, then θ_{SS} would have to be between roughly 0.2 and 0.4. It should therefore be clear that SSD hyper parameters would have to be tuned simultaneously, since certain settings may no longer be applicable once the other is varied.

Secondly, the fundamental premise of this SSD technique is to tune the hyper parameters such that transient and quasi steady states become distinguishable by means of a threshold (θ_{ss}). This threshold can range from 0 to 1, thus ideally the greater the difference between the stationarity (estimated

probability of stationarity) of the actual transient and the actual quasi-steady, the more effectively the SSD will operate. Figure 17 clearly shows that at small window sizes (below 100 samples) the actual transient and steady probability of being stationary is very similar, thus making it difficult to distinguish these different states by means of a threshold. Figure 17 shows that as the window size increases the difference between the actual transient and stationary becomes larger, thus allowing for more leniency in setting the threshold. The larger the window size, the less prone the SSD technique is to process noise, allowing the signal to reach some level of stability (Kelly and Hedengren, 2013).

However, Figure 17 also shows that as the window size continues to increase, the distinguishability of actual transient and stationarity reduces. At a window size of 1600 samples, the transient and stationary states are once again not distinguishable. This can be explained by the fact that too long of a window size will reduce the impact of a short transition period, therefore resulting in the window seeming stationary. Transient and steady data therefore become indistinguishable, which are well known effects termed aliasing (Kelly and Hedengren, 2013). It should be noted that the mean stationarity of the actual stationary states in Figure 17 remains relatively constant at larger window sizes. The dataset mainly consists of stationary states, thus the inclusion of transient data within the analyses has minor effects on the overall mean of the actual stationary states.

Although the mean stationarity of actual transient and steady states effectively convey the effects of varying window size, these metrics are not very useful for evaluating the overall performance of SSD tuning and technique. This is due to the fact that these metrics do not take into account the SSD model sensitivity to the threshold θ_{SS}. A useful technique used to compare various fault detection models is known as the Receiver Operating Characteristic (ROC) plot. The ROC curve can be used to depict the trade-off between fault detection and false alarm rates by means of determining the area under the curve (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019). The higher the area under the curve (AUC), the better the performance of the monitoring approach. The ROC curve however only becomes useful when the training data contains faults, allowing for an optimal threshold to be chosen depending on the goals of the monitoring approach (Addo, 2019).

The ROC curve is thus ideal for the evaluation and comparison of SSD results and tunings. The ROC curve obtained from the default tuning parameters described in Table 9 (with the exclusion of θ_{SS}, which is varied) is displayed in Figure 18.

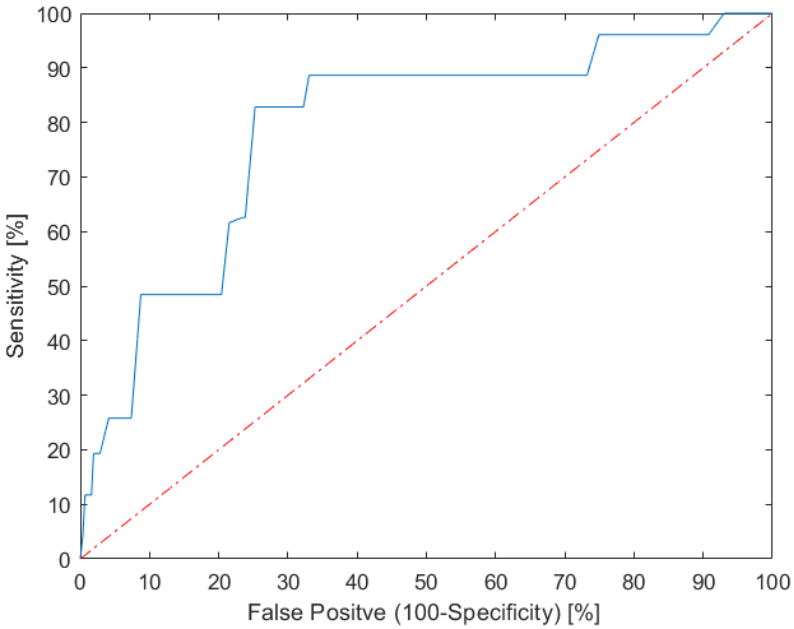


Figure 18: ROC curve obtained from default SSD settings described in Table 9 with an AUC of 0.79, here the dashed line indicates the chance diagonal and the solid line indicates the SSD performance at 0.01 θ_{SS} intervals

The sensitivity seen in Figure 18 can be seen as the true transient rate $(1 - \mathrm{MTR})$ of the SSD model, whereas the false positives (FTR) can be related to the specificity of the SSD model. Figure 18 shows that the sensitivity of a SSD technique is intrinsically linked to the specificity of the technique. When the SSD threshold is varied usually both sensitivity and specificity vary, however in different directions. As a result sensitivity and specificity have to be considered together, which can make the analyses difficult (Halligan, Altman and Mallett, 2015).

The AUC is thus an ideal measure that combines the specificity and sensitivity into a single diagnostic measure. The need to "juggle" between specificity and sensitivity is thus not required (Halligan, Altman and Mallett, 2015). The dashed red line seen in Figure 18 is known as the chance diagonal, which has an AUC of 0.5. If a SSD monitoring model has an AUC larger than 0.5, it detects transients more effectively than if they were detected with no discrimination.

AUC is certainly useful, the metric does however have certain issues. Firstly, certain ROC curves may have the same AUC, but one would be more preferable than the other, the AUC metric does not contain this information. Further, if AUC is used as an evaluation metric, it is assumed that specificity and sensitivity are of equal importance. The use of SSD in this investigation is to filter out transient data, such that steady states can be analysed effectively as described in 3.2.3. For the dataset described in Table 8, steady states compromise 98 % of the entire dataset. Figure 18 shows that at a certain SSD threshold (θ_{SS} at 0.33) 88 % of the transients are detected and can be removed, however 33 % of the steady sates will be removed along with the transient data. Since the steady state data forms most of the dataset, this will result in roughly 33 % of the entire dataset being removed. If the dataset was for example made up of a single mode and various transients, this would not be such a drastic issue. However as described in Table 8, the

dataset was composed of 6 modes. An entire mode could have been removed, resulting in essential information being lost from the dataset. The AUC fails to display this information.

Keeping AUC issues in mind a hyper parameter investigation of varying the window size, significance and variables considered can be seen in Table 10. Table 10 columns are made up of different window sizes (number of samples) and the rows at different significance levels. A different set of variables is considered for each different block of Table 10. Here 2 PCs denotes that the first two principle components were considered within the SSD analyses. The coloured cells display the different AUC results obtained from the different SSD parameter settings.

Table 10: SSD hyper parameter investigation on dataset described in Table 8

Sig (α) ↓	Only Cooling Water Window Size (*n*) →			All MVs and CVs Window Size (*n*)			All Variables Window Size (*n*)		
	100	***300***	***500***	***100***	***300***	***500***	***100***	***300***	***500***
0.01	0.627	0.603	0.580	0.798	0.863	**0.877**	0.824	0.888	0.880
0.05	0.721	0.729	0.714	0.742	0.847	0.745	0.772	0.896	0.777
0.1	0.714	**0.769**	0.677	0.702	0.824	0.670	0.746	**0.901**	0.711
	1 PC			2 PCs			3 PCs		
	100	**300**	**500**	**100**	**300**	**500**	**100**	**300**	**500**
0.01	0.304	0.632	0.629	0.543	0.721	0.701	0.630	0.731	0.745
0.05	0.700	**0.794**	0.721	0.724	**0.763**	0.661	0.746	**0.768**	0.697
0.1	0.745	0.773	0.661	0.718	0.740	0.619	0.727	0.763	0.675
	4 PCs			5 PCs			6 PCs		
	100	***300***	***500***	***100***	***300***	***500***	***100***	***300***	***500***
0.01	0.780	0.774	**0.875**	0.742	0.727	**0.841**	0.731	0.721	**0.827**
0.05	0.784	0.841	0.802	0.771	0.782	0.708	0.761	0.760	0.661
0.1	0.749	0.826	0.673	0.730	0.753	0.562	0.733	0.724	0.509

Table 10 shows that all SSD tunings resulted in AUC results that performed better than chance (>0.5), except for the setting where only one PC was considered, with a window size of 100 samples and a significance of 1 %. At this setting, the SSD technique seems to switch transients with steady states. Table 10 also shows that analyses of cooling water or the first PC only results in the worst results in overall. It can thus be concluded that key information of transients is lost if the other variables are not considered.

Surprisingly, analyses of all variables resulted in the best AUC. Even better than if only the manipulated and controlled variables are considered, as seen in Table 10. This result is surprising due to the fact that most of the process variables of the CSTR are correlated to the manipulated and controlled variables. Thus, it is expected that consideration of solely these variables would be sufficient (Kelly and Hedengren, 2013). For this dataset however it is clear that at the specified significance levels, consideration of only the manipulated and controlled variables, would result in a SSD tuning that is not sensitive or specific enough to transients, therefore with the inclusion of all other variables the AUC can be improved.

In terms of window size and significance, it seems that the choice of these is highly dependent on the variables considered and vice versa. In general it seems that the larger window size with a lower

significance performs SSD more effectively. However, from Table 10 it is clear that for the chosen SSD technique the hyper parameters have to be set simultaneously, thus obtaining an effective SSD analysis is quite difficult.

Further, Table 10 shows that the analysis of four PCs resulted in an AUC of 0.875, at a window size of 500 samples and 1 % significance. The consideration of PCs within SSD is thus validated and may achieve similar performance to when all variables are considered. It is also clear that from one to four PCs the AUC improves in overall, however with the inclusion of more PCs the AUC diminishes. It can therefore be concluded that the first four PCs contain enough information for SSD to be performed effectively. Figure 19 corresponds quite well to the results shown in Table 10. As discussed in 2.3.2.1 various methods of selecting PCs exist. One of these methods is known as the scree test, which states that the dimension of the PC space should be set to the dimension where the variance explained profile is no longer linear (Chiang and Russell, 2001).

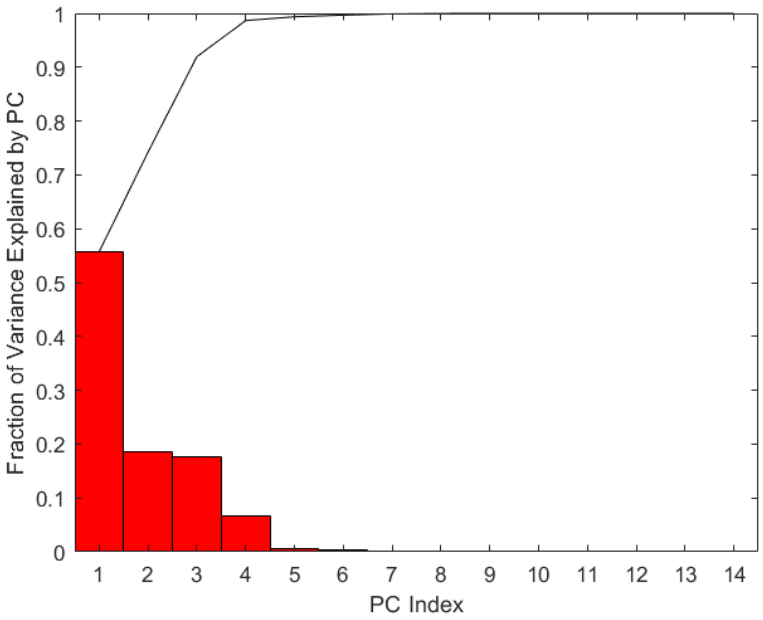


Figure 19: Pareto Chart of the dataset described in 4.1.1 displaying how the PCs explain the variance

Since the variables of the CSTR system are highly correlated, Figure 19 firstly shows that the dimensionality of the CSTR data can be explained to a great extent by only a few PCs. The bars of Figure 19 however do not display a single “elbow” (no longer linear profile), making the scree test challenging (Addo, 2019). Upon subjective analysis however, taking into account the cumulative variance explained and the “elbows”, it is clear that the selection of the first four PCs would be sufficient, describing 98 % of the variance.

Further, similar to the issue of testing multiple univariate control charts, performing multiple t-tests may reduce the accuracy of SSD technique (Xu, Wu and Tseng, 2018). This phenomena can clearly be seen in Table 10 when analysing the PC results. Based on Figure 19, PCs five to fourteen explain no useful

information, the inclusion of these within the SSD analysis will increase the number of t-tests that need to be performed, decreasing the AUC, as seen in Table 10.

Analysis of Figure 17 and the results from Table 10, it is clear that the SSD algorithm is not robust to window size. The data partitioned into windows method as described in section 3.2.2, results in erratic behaviour in the results, best seen in Figure 17. The effectiveness of the algorithm very much depends on the manner the window "falls" on the data. Further tuning of the other SSD parameters can then be implemented such that transient detection becomes effective at a specific window size, however considerable knowledge of the ground truth would be crucial for this. The unpredictable nature of the algorithm is a major drawback of the explained SSD technique.

Key outcomes of SSD hyper parameter tuning can therefore be summarised as follows:

1. Considerable ground truth of quasi steady and transient states must be known
2. Varying process dynamics, such as different time constants of transitions, short quasi steady periods and various modes may reduce the effectiveness of a single SSD tuning. Thus setting the window size according to the time constants of transitions is a good starting point, but may oversimplify the issue.
3. The window size is an extremely important hyper parameter which is difficult to set. a) Due to the erratic nature in effectiveness resulting from the manner the data is segmented into windows, b) the window may not be too large (insensitive to transients) or small (sensitive to process noise).
4. Performing SSD on PCs may ease tuning, especially for processes with many more variables.
5. The scree test may be an effective method of selecting the number of PCs to perform SSD on.
6. The hyper-parameters are highly interlinked and have to be tuned simultaneously.
7. SSD tuning is specific to the process data. Similar to the "No Free Lunch Theorem" (discussed in 2.4.1), a SSD tuning may work on a specific set of data, but could completely fail on another. This may even occur within the same dataset (refer to point 2.).

4.1.2.3 Best SSD tuning

Based on the ROC curve and its AUC, a preferred SSD tuning is described in Table 11, achieving an AUC of 0.91 (by varying the SSD threshold). The SSD tuning described in Table 11 resulted in all transients being detected (MTR of 0 %), however also resulted in a 24 % FTR. This tuning was selected over various trailed configurations and chosen since it achieved a low MTR with a relatively low FTR (ie. based on ROC). It should be noted that this selection procedure can only be performed since the ground truth of the process state is known. In an industrial setting this cannot be done.

Table 11: Adequate parameter settings for SSD

Window Size (*n*)	**Variables**	**Threshold (θ_{ss})**	**Significance (α)**
700	All Variables	0.7	0.01

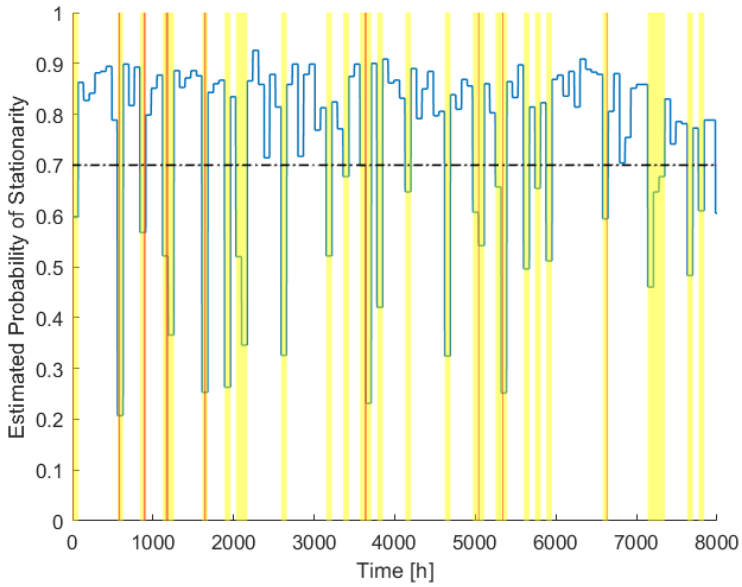


Figure 20: SSD results obtained from default tuning, where yellow and red shaded regions denote detected and actual transient periods. The blue data points describe the probability of a time window being stationary and the dotted line denotes Θ_{ss}

The results are visually displayed in Figure 20, where it is clear that all transients (red regions) were detected (yellow regions). Thus resulting in the 0 % MTR. Figure 20 however also shows that the detected transients are far longer than the actual transients, as well as the fact that certain transients were detected where no actual transition occurred, thus the high FTR.

As discussed with the issues of AUC, the determination of an effective SSD threshold is difficult. The role of SSD in this investigation is to filter out all transients, since their effect may be detrimental to subsequent steps. Thus ideally, a MTR of 0 % would be required. However determining an acceptable FTR is an issue. In this case, it was decided that a FTR is acceptable when the majority of the quasi steady state data structure remains within the filtered dataset. In other words, all modes occurring within the original data must be somewhat present in the filtered data. This was the case at the chosen threshold, as can be seen in Figure 21.

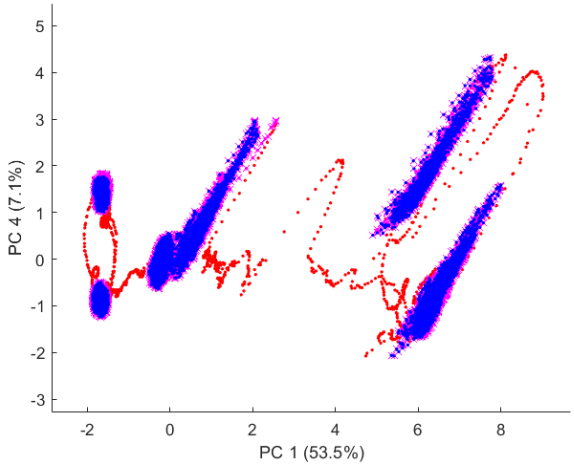


Figure 21: Multimodal CSTR data in the PC space where red and blue dots display the filtered and unfiltered data. The magenta crosses represent the true steady state data

Modes or quasi steady states are concentrated in small ellipses, in contrast transitions are scattered over a larger area in the latent space (Srinivasan, Wang and Ho, 2004). PCs 1 and 4 effectively displayed all 6 modes separately, thus served as an ideal sub space to visualize SSD performance. Figure 21 clearly shows these effects, as well as the fact that the detected steady states visually correspond well to the “true” steady states which were determined as explained in section 3.1.2.

It can therefore be concluded that in this case a false transient detection rate or FTR of 24 % sufficiently maintained the structure of the CSTR data (within the considered dimensions). Further, it can be seen that the “true” steady state did not always conform to the statement that modes are concentrated in small ellipses. This can be seen in Figure 21 and more clearly in Figure 22, where the magenta crosses (“true” quasi-steady state) over lay the red dots (detected transients), in the latent space it is clear that these data points should rather be deemed as transient data. The specific transition discussed here resulted due to decrease in methanol inlet molar flowrate, effects of which propagate to the entirety of the CSTR system. Thus even minor changes in inlet molar flowrate will have considerable effect on the overall system, making the “true” SSD determination technique prone to error.

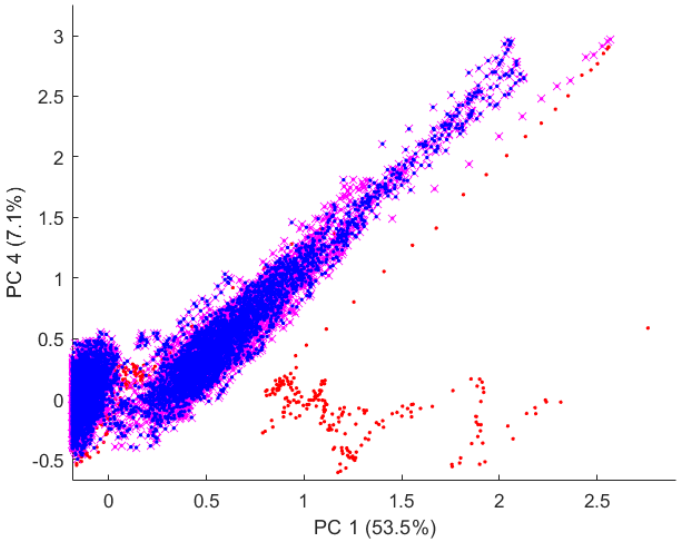


Figure 22: Magnified view of Figure 20, where red and blue dots display the filtered and unfiltered data. The magenta crosses represent the “true” steady state data

It should therefore be clear that obtaining the “true” steady state even from noise free data is quite a difficult task in itself and in a sense is somewhat subjective (procedure 3.1.2). One should thus be careful when performing SSD and filtering data, that the “structure” of the data is not manipulated to an extent where it is of no use.

4.1.2.4 Effect of sample rate on SSD

Further, as described in 3.2.2 and 3.3.1, SSD in this investigation is applied to historic data only. This historic data may however only be maintained at a lower sample rate, for example due to storage concerns. The sample rate of data will have an effect on the effectiveness of SSD. This effect can be seen in Table 12. It should be noted that the window time duration investigated is the same as in Table 10, however since the sample rate is reduced, the window size in samples (n) is reduced by a factor of 10.

Table 12: SSD hyper parameter investigation on dataset described in Table 8, however at a sample rate of 1 sample per hour

Sig (α) ↓	Only Cooling Water Window Size (n) →			All MVs and CVs Window Size (n)			All Variables Window Size (n)		
	10	***30***	***50***	***10***	***30***	***50***	***10***	***30***	***50***
0.01	0.280	0.465	0.485	0.468	0.812	**0.874**	0.549	0.875	0.884
0.05	0.550	0.716	0.729	0.714	0.834	0.742	0.739	**0.882**	0.768
0.1	0.673	**0.776**	0.676	0.708	0.801	0.656	0.713	0.874	0.662
	1 PC			2 PCs			3 PCs		
	10	***30***	***50***	***10***	***30***	***50***	***10***	***30***	***50***
0.01	0.067	0.324	0.430	0.134	0.511	0.665	0.150	0.594	0.773
0.05	0.407	0.797	0.684	0.611	**0.789**	0.596	0.683	0.810	0.710
0.1	0.724	**0.807**	0.671	0.717	0.770	0.619	0.718	**0.854**	0.745
	4 PCs			5 PCs			6 PCs		
	10	***30***	***50***	***10***	***30***	***50***	***10***	***30***	***50***
0.01	0.224	0.672	**0.871**	0.296	0.659	**0.854**	0.314	0.657	**0.829**
0.05	0.732	0.866	0.809	0.710	0.774	0.674	0.668	0.738	0.662
0.1	0.739	0.848	0.696	0.690	0.750	0.544	0.634	0.727	0.508

Comparing the results from Table 12 and Table 10, it can be seen that the overall performance of SSD at a higher sample rate (6 samples per hour) is better (in terms of AUC). The degradation in performance is however not drastic, thus gives a good indication that SSD could be performed on datasets where the historic data has been reduced in sample rate. A reason for the somewhat maintained performance may be due to the fact that autocorrelation within the data could have been removed to a certain extent (as seen in Figure 14). Autocorrelation is not accounted for in the test statistic (in this case the t-test) and may thus negatively impact the results (Rhinehart, 2013). However, the reduced sample size or window size (not window time duration) increases uncertainty, which in this case has a more detrimental effect on the SSD performance.

Within this CSTR simulation, transients are preceded and succeeded by relatively long quasi steady states or modes. This is key property of continuous processes data as a result of the control objectives, specifically to allow for smooth operation and production. Altering the sample rate of process data such that transient data is minimized could be a valid approach for many continuous processes. An example of this method can be seen in Figure 23.

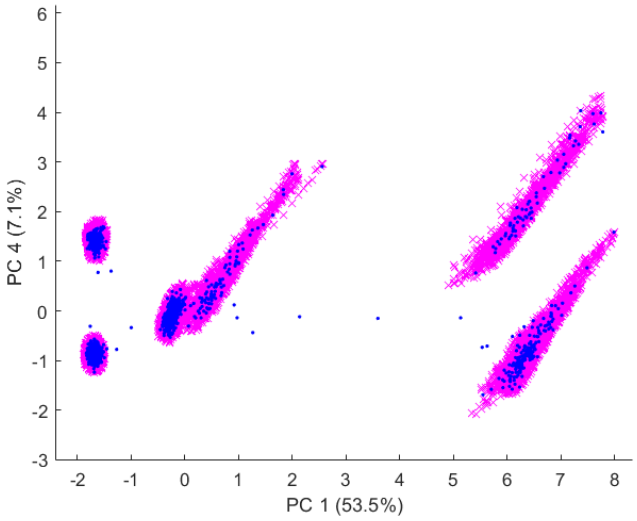


Figure 23: Blue dots represent the subsampled data (sampled every 5 hours) and the magenta crosses describe the "true" quasi steady data (normal sample rate) in the latent space

Comparing Figure 21 and Figure 23, it is clear that the transient data has effectively been reduced in this manner without the need of a complex algorithm. As a result, subsampling data may be a useful alternative to SSD. However, even though the CSTR data was sub sampled to a sample every five hours, certain transient data still remained. Cases of this can be seen in Figure 23 where the blue dots are not accompanied by magenta crosses. Further, it is clear that important mode information is lost when data is subsampled. Subsampling data results in the covariance structures of the various modes being altered to a certain extent, most likely due to the reduction in autocorrelation within the data. This was not the case in Figure 21. SSD is therefore more difficult to tune and apply to data, but may be more effective in preserving the structure of the data.

4.1.2.5 Sliding Window SSD

The randomness in performance of the SSD algorithm (3.2.2) resulting from data being partitioned into windows is a major issue, which further thwarts effective SSD tuning. A more robust approach to SSD would therefore be to adjust the SSD algorithm seen in 3.2.2 such that it is applied with a sliding window. The code of which can be found https://github.com/FrancoisNoelle/DecisionSupport. The effect of "chance" will thus be reduced, allowing for improved tuneability, which is a difficult task in itself. Further, the adjusted SSD technique will become more viable for real time or online implementation. The implementation of the threshold SSD technique with a sliding window has not yet been discussed in literature. The tuning parameter settings for this investigation can be seen in Table 13.

Table 13: Sliding window SSD parameters settings

Window Size (*n*)	**Variables**	**Threshold (θ_{ss})**	**Significance (α)**	**Assignement Delay**
700	All Variables	0.3	0.01	0

As seen in Table 13 an additional tuning variable is required. The assignment delay hyper-parameter specifies to which sample within the window (n_i) the estimated probability of stationarity of the entire window is assigned to. Therefore, an assignment delay of 0 would result in the final sample being assigned the estimated probability of stationarity. The visual results at this tuning can be seen in Figure 24.

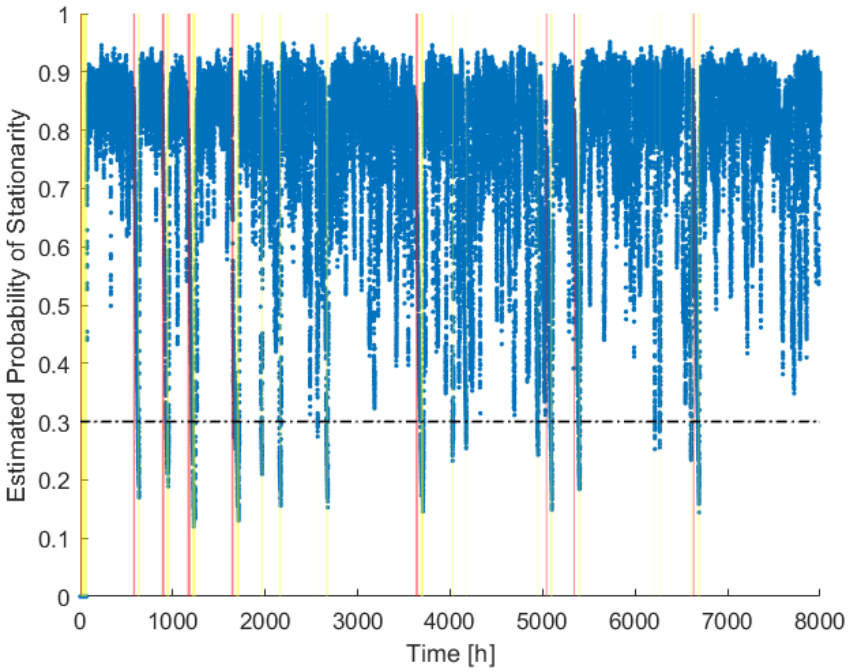


Figure 24: Sliding Window SSD results obtained from tuning in Table 13, where yellow and red shaded regions denote detected and actual transient periods. The blue data points describe the probability of a time window being stationary and the dotted line denotes Θ_{ss}

Firstly, a transient period from samples 0 to 699 occurs as result of the nature of the algorithm. In this case however, this is not too much of an issue, since the initial data is transient as a result of the CSTR start-up. Secondly, it seems that at this tuning, all instances, the yellow (detected) and red ("true" quasi steady states) shaded regions correspond. Thirdly, it is clear that the stationary fraction results seen in Figure 24 do not exhibit the "stepped" results seen in Figure 20, due to all samples within the dataset being assigned a steadiness value. Further, only infrequently and for short periods does false transient detection occur without being in the vicinity of an actual transient.

Figure 25 more clearly displays some of the results seen in Figure 24. Here however it is clear that the detected transients succeed the actual transients. As a result the performance of the tuning is quite poor in terms of FAR and TAR. This is mostly due to the fact that the delay is tuned improperly, a better set delay could result in better SSD performance. This delay will have to be adjusted with window size, larger window sizes will require a larger assignment delay. In Figure 25 for example, the transient occurring at around 3600 hours was consistently detected 56 hours too late. Here consistent refers to a considerable amount of transient samples in consecutive order.

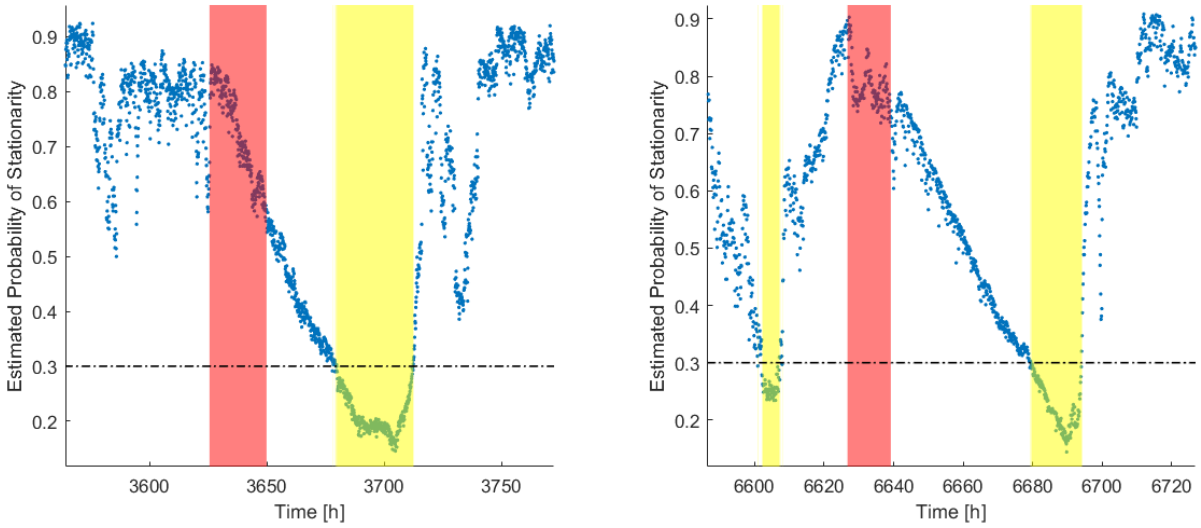


Figure 25: Magnified views of Figure 24, displaying the true (red) and detected (yellow) transient periods

Comparing Figure 16 to Figure 25 it can also be seen that the detection resolution of the sliding window (actual transient duration is similar to detected transient duration) outperforms that of the data partitioning method, however at a delay. If this technique is therefore tuned properly a lower FTR can be expected. Including a tuning parameter denoting a minimum transient period could be effective in further reducing FTR (similar to an alarm detection delay). The adjusted SSD algorithm becomes more robust in overall, but at the expense of an additional tuning parameter. Setting the assignment delay hyper-parameter is also a major concern addressed in literature (Xu, Wu and Tseng, 2018) and crucial for online SSD implementation.

For offline implementation however this additional hyper parameter makes filtering transient data especially difficult, since the alignment of the detected transient and the true transient must be achieved effectively. Due to varying process dynamics occurring as result of various transitions it is not an easy task aligning this data effectively. Further, varying the SSD threshold (Θ_{SSD}) will also affect the detection resolution. The partitioning window SSD method (3.2.2) achieves satisfactory results without the need of tuning this additional hyper parameter.

4.1.2.6 Concluding remarks

The various steady state analysis configurations are effectively able to detect transients, however certain concerns of the technique have to be addressed. Firstly, in industry transient or mode switching operations are usually more complex, unlike with the CSTR data. Operators usually wait for the effect of one set of actions to stabilize before proceeding to the next set, therefore multi-step procedural operation manifests as an alternating sequence of transient and quasi steady behaviour (Srinivasan, Viswanathan and Vedam, 2005). From the investigation, it is clear that the window size (in time duration) is required to be considerably long, to increase the SSD techniques robustness to process noise. As a result these short modes or quasi steady periods will most likely be deemed transient, thus filtered from the data for subsequent procedures. Similar issues may be experienced when the subsequent mode is not much longer (follow-up transition occurs quickly) than the transition itself. Fault or abnormal data

will most likely also not manifest into a quasi-steady state since operator intervention will usually occur prior to this (Liu and Chen, 2010), and as a result will most likely be filtered from the dataset. Depending on the goals of the subsequent analyses this may be advantageous or disadvantageous.

Further, industrial processes most likely have higher dimensions, various time constants and more complex non-linear interactions, all of which may pose challenges to the described SSD algorithms. Long process settling times and time delays may pose serious challenges to the effective implementation of SSD as well (Chen and Howell, 2001). Plant-wide stationary processes are hard to meet within the industrial context, the steady state requirement is quite demanding (Kruger and Xie, 2012). Different plant sections can undergo state changes independently, hierarchical division of plant units and their process data is therefore essential before analysis is performed (Srinivasan, Viswanathan and Vedam, 2005). Effective SSD implementation therefore requires the integration of expert knowledge, both for data segmentation and tuning.

However once SSD is tuned and applied effectively, it poses various benefits. For example, SSD has been implemented in a fully integrated oil refinery, where its results serve as a key process performance indicator (Kelly and Hedengren, 2013). SSD has also been implemented for data reconciliation and process optimisation, where process models require the system to be steady (Mansour and Ellis, 2008). In a sense SSD can also be implemented for fault detection itself and assist in the isolation of temporal root causes of process incidents, even if the data is multimodal (Chen and Howell, 2001). Further, SSD has for example been used to identify operating points in engine flight data, which could then be used for condition monitoring purposes (Simon and Litt, 2011). Finally, SSD is a useful tool for data pre-processing or filtering as the case in this investigation, which is able to effectively maintain the quasi steady state “structure” of process data. The discovery and identification of modes using clustering algorithms may be more effective once all or most transients have been removed from the process data.

4.2 State Based Analysis Applied in a Supervised Approach

4.2.1 The need for stationarity analysis and GMM

The following investigation will be conducted on the CSTR simulation dataset described in 4.1.1. As mentioned in 2.6.1 and 2.6.2, in literature it is often assumed that process data does not contain transients, in industry however this is usually not the case (Thomas, Zhu and Romagnoli, 2018). The investigation that that follows will therefore discuss the effects of transient data on the suggested procedure described in 3.2.3.

4.2.1.1 Variable Selection

As discussed in the 3.2.1 and 2.4.2.3, the dimensionality of the CSTR data is first reduced to address issues of overfitting and the curse of dimensionality by means of PCA. Dimensionality is however also reduced to assist in data visualization such that experts or human agents can effectively analyse the process data. Ideally, all key data features should be presentable within three dimensions.

Figure 26 shows the CSTR data from 4.1.1 projected into various PC spaces, where the various colours indicate the different states. It should be noted that the dark blue data points represent transient states, all of which are grouped into a single state. The various transitions are therefore not distinguished. The remaining colours distinguish the different modes or quasi-steady states. Figure 26 d) is most effective in displaying the six different modes (information from Table 8), this PC space is therefore useful to visualize results from the various procedures (SSD, K-means clustering, GMM).

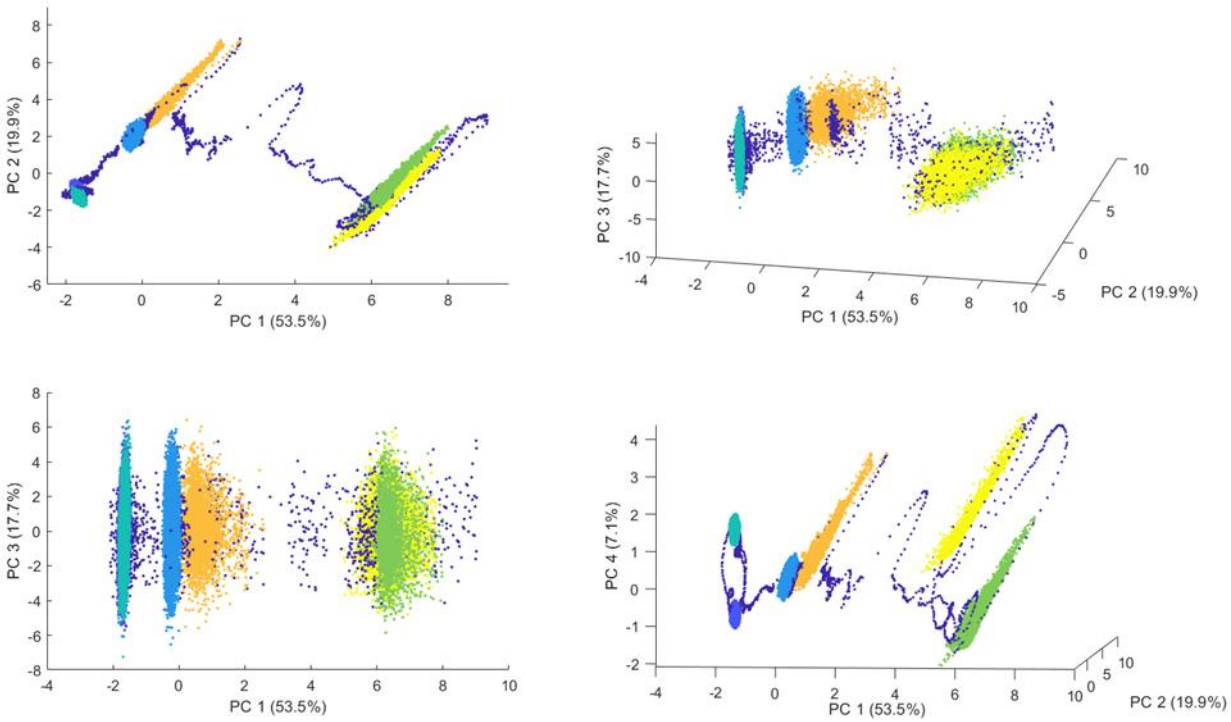


Figure 26: Visualizing CSTR simulation data in various latent spaces where the various colors represent the various states (including transient) a) 4 Modes visible b) 4 or 5 modes visible, however not well separated c) 4 modes visible d) All six modes are clearly visible and well separated

It should be noted that the start-up period is removed from Figure 26 to improve the interpretability of the modes in the PC space (not removed during training). The PC space including start-up can be seen in Appendix A Figure 54. Further, the GMM training and testing procedure can be implemented on higher dimensions, however for this section of the investigation it is crucial for the results to be visually interpretable. It was therefore decided to conduct further investigation considering principal components 1 and 4 only, as seen in Figure 27. This investigation is therefore performed in a supervised manner, selecting only variables that best separate the 6 modes within as few dimensions as possible. The unsupervised approach will be discussed later.

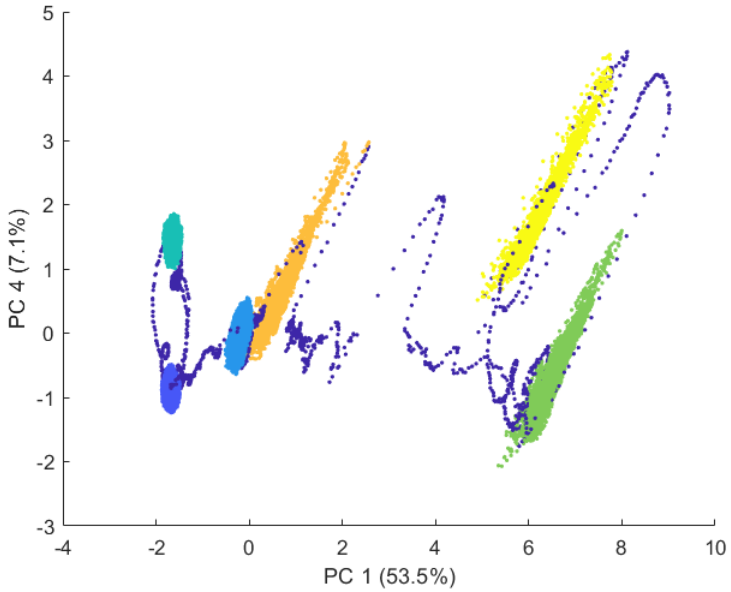


Figure 27: Visually Interpretable Latent Space Distinguishing the various Modes

Comparing Figure 27 and Figure 21 it should be realised that both these figures are the same, however in Figure 27 the various modes are distinguished by means of the input variable pairings and the “true” SSD results obtained from the noise free CSTR simulation. In other words, Figure 27 represents the ground truth, obtained as discussed in the 3.1.2.

4.2.1.2 Analysis via K-means

Often it is considered to be that modes can be effectively monitored/detected by simply comparing the distance of a data point to the multivariate mean of a mode (Srinivasan, Wang and Ho, 2004). This investigation is performed to determine if such a detection/monitoring scheme is effective on the CSTR simulation data.

For the purposes of this investigation, it is assumed that the number of modes is known prior to the parameter fitting. As shown in Table 8, the CSTR dataset contains 6 modes. Here it is decided to make use of K-means clustering, since it separates modes/clusters based on the Euclidean distance of data points from cluster centres/means. It should further be noted, that the K-means algorithm needs a method of initialization, in this investigation the K-means clustering algorithm, described by Arthur (2007), was used to initialize clustering. The clustering results achieved for both the filtered steady state

(transients removed using SSD tuning described in Table 11) and unfiltered CSTR process data (no detected transients removed) can be seen in Figure 28.

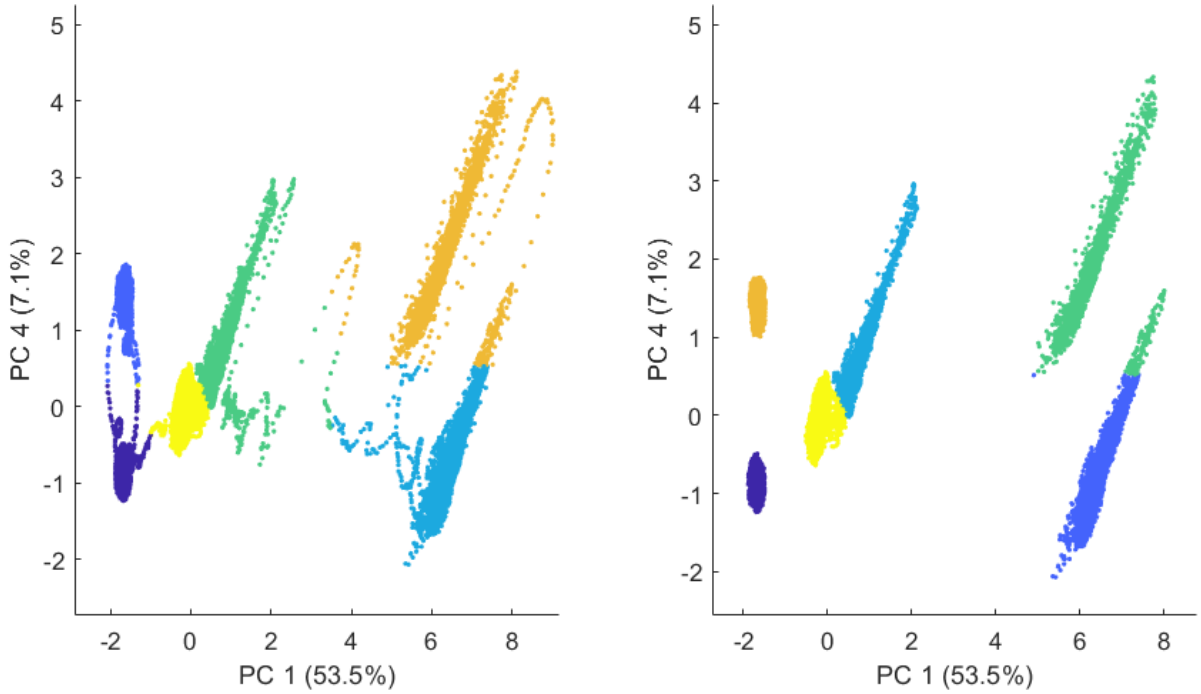


Figure 28: Clustering Results obtained with K-Means Algorithm where the 1st and 4th PCs are considered with 6 clusters fit to the CSTR Data, Crosses denote the cluster means a) All CSTR Data Clustered b) Only Stationary Data Clustered

Comparing the true modes seen in Figure 27 with the detected modes seen in Figure 28, K-means clustering is reasonably effective at segmenting the modes. From Figure 28 a) however it can be seen that the transient data is not distinguished from the modes, rather transient data is assigned to various modes, as a result of the hard cluster assignments. The K-means algorithm attempts to fit data into hyper-spherical clusters, therefore resulting in the long, narrow transitions between steady states being assigned with modes (Thomas, Zhu and Romagnoli, 2018).

Figure 28 b) visually reflects the true clustering seen in Figure 27 more accurately, with the exclusion of transient data. It should be noted that the same modes are described by different colours Figure 27 and Figure 28 (even a) and b), as a result of varying mode indexing (mode identifying indices). This is an important issue that has to be considered, especially in literature. Since the ground truth is known here, it has to be ensured that the “detected” states and the “true” states have the same identifying indices. Only then can the performance of the monitoring scheme be evaluated effectively as discussed in 3.3.2. Since the K-means clustering algorithm initializes the first cluster (index 1) at random, the indices are in a sense randomly assigned. The need for the state identifying scheme described in section 3.1.2 is thus validated, identifying modes based on their weight or fraction of data.

Since Figure 28 b) achieves good visual clustering performance, it may be thought that an effective monitoring model can be setup simply by using purely K-means clustering (especially when transients are removed prior). However as seen in Figure 29, this is not the case.

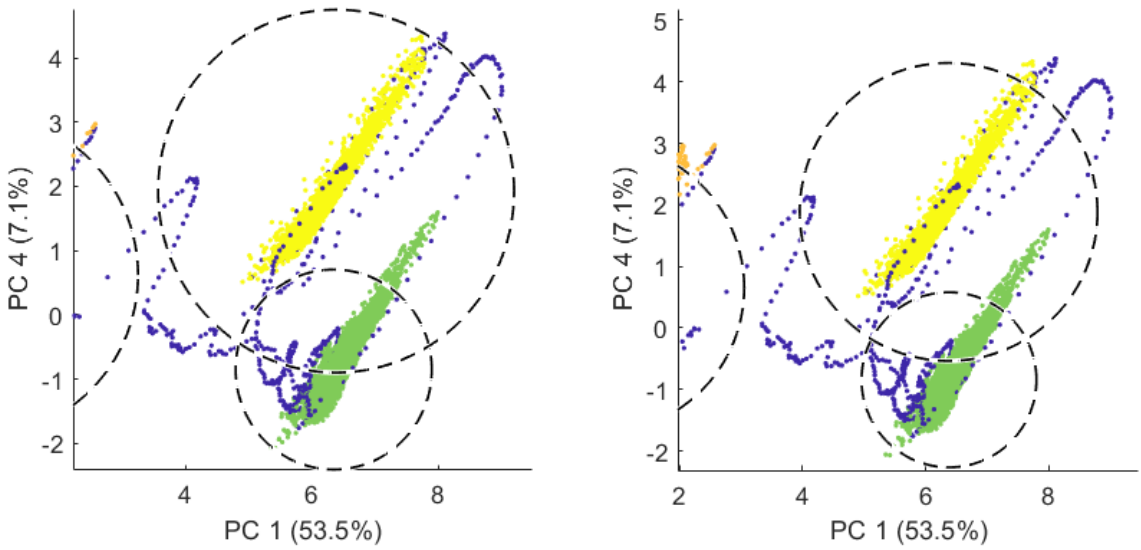


Figure 29: Determined Mode Thresholds displayed on magnified true Mode and Transients in Latent Space a) 99th percentile of Euclidean Distance Training Data Determined from K-means clustering where transients were not removed b) 99th percentile of Euclidean Distance Training Data Determined from K-means clustering on stationary data only

Figure 29 shows that detection thresholds for both a) and b) are "spherical", as a result of the K-means algorithm. Since K-means assigns data to clusters/modes based on their Euclidean distance from the cluster centres, the detection thresholds will also be based on the Euclidean distance. Thus, it is assumed that cluster/mode variance is equal in both latent dimensions, which is clearly not the case in both the displayed modes. As a result thresholds determined from K-means clustering may be predisposed to similar issues described with univariate detection thresholds described in 2.3.2.2. The detection boundaries do not describe the true modes well in both Figure 29 a) and b). Both transient data (blue data points) are contained and true mode data (green/yellow) are not contained within the detection boundary. Transient data will thus often be falsely identified as being part of the modes, as well as the modes being mistaken for each other. Further, it is clear that even if the covariance of a mode is effectively modelled, certain transitions will still not be distinguished from the modes within the chosen dimensions.

Comparing Figure 29 a) and b), it can be seen that no clear differences in in the detection boundaries exist. Figure 29 a) however has larger confidence boundaries around the modes compared to b). From Figure 29 however it is unclear which monitoring scheme would result in better performance (balancing false positives and false negatives). From this investigation it is clear that K-means clustering has issues from the monitoring perspective, mostly contributed due to the facts that data points have hard cluster assignments and that the covariance of modes is not effectively modelled. The K-means algorithm however remains important since it is able to somewhat effectively separate mode data as seen in Figure 28 without extensive computation.

4.2.1.3 Implementing GMM

The use of an alternative multimodal monitoring/detecting technique is thus validated. In this investigation GMM is the chosen technique. Similar to K-means certain initialization parameters are required prior to fitting the GMM to the data. As mentioned in 2.4.2.3, it is crucial for the initialization parameter estimates (Θ_0) to be as accurate as possible to prevent the EM algorithm (2.4.2.1) from converging to a local maximum or not converging at all. This is however a difficult task due to the unsupervised manner the data has to be dealt with. Since the number of actual modes or quasi steady states within the data is usually unknown, it further complicates initial parameter estimation. For this investigation however it is assumed that the number of modes is known (6 modes), as in the K-means investigation in 4.2.1.2.

Various methods exist for selection of these initialization parameters, for example initialization parameters can be set randomly. Most notably K-means clustering discussed in 2.4.1 has been widely implemented to estimate the EM initialization parameters (Yu, 2011). Due to the effectiveness and simplicity of the algorithm as determined in 4.2.1.2.

Visual results of the GMM fit on the CSTR dataset containing both stationary and transient data are shown in Figure 30. Here the EM algorithm was initialised with parameter estimates obtained from K-means clustering, which was replicated 10 times with different seeds. The K-means cluster analysis with the lowest sum of squared distances was chosen as the seed (Θ_0) for the GMM fitting procedure. Further, the GMM covariance was set to full, thus is not constrained in any analysed dimension. The covariance was also not tied, therefore a different covariance could be fit to each mode. All in all 36 parameters have to be fit to the CSTR data via EM.

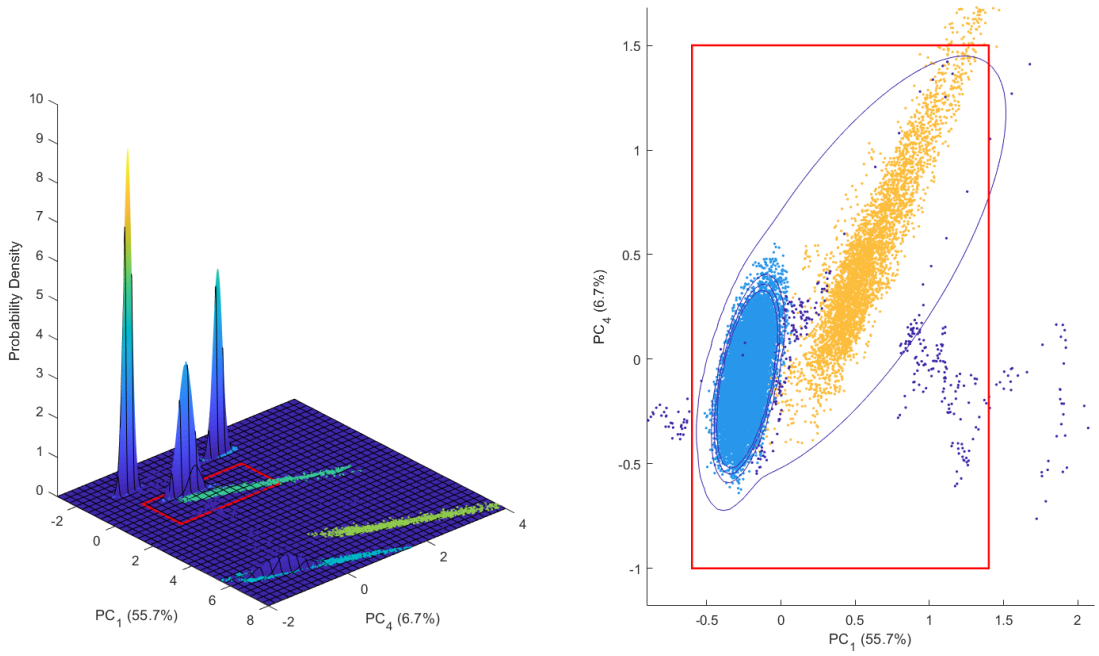


Figure 30: a) Surface Plot of the GMM fit on the CSTR data containing transitions b) Red bounding box in (a) magnified as a contour plot with probability density level intervals [0.01:0.05:0.2]

Figure 30 a) shows four peaks at the multivariate means of the true modes. Visually it is clear that Gaussians were effectively fit to these modes (the height of the peaks however is not indicative of the fit accuracy, but rather the variance of the modes). The dataset however contained 6 modes, Figure 30 a) fails to display these modes. The probability density of the remaining two modes is so low (relative to the other four) that they do not appear on Figure 30 a). Figure 30 b) displays a magnified contour plot of the red bounding box in Figure 30 a). Figure 30 b) shows that Gaussians are fit to both modes. However it can be seen that the covariance of one of the Gaussians (orange) does not accurately describe the "true" covariance of the mode. Similar to the K-means, transient data tends to stretch the covariance in various dimensions since all data points are considered during EM convergence. Therefore the EM algorithm is more likely to converge to a local maximum, not effectively describing the true parameters of the multimodal data. Figure 30 however also shows that certain modes are more prone to these issues than others. Unlike the detection boundary achieved with K-means which is theoretically constrained to being hyper-spherical, GMMs are able to effectively take into account the covariance of the various modes.

Comparing contour plots in Figure 31 b) and Figure 30 b), it can be seen that when transient data is removed prior to fitting, the overall parameter estimation by means of EM is improved. The stretching effect (tendency to increase eigenvalues of covariance matrix) transient data has on the covariance matrix is avoided. The initialisation parameters obtained via K-means are also closer to the true parameters, and as a result EM procedure is less likely to converge on a local maximum.

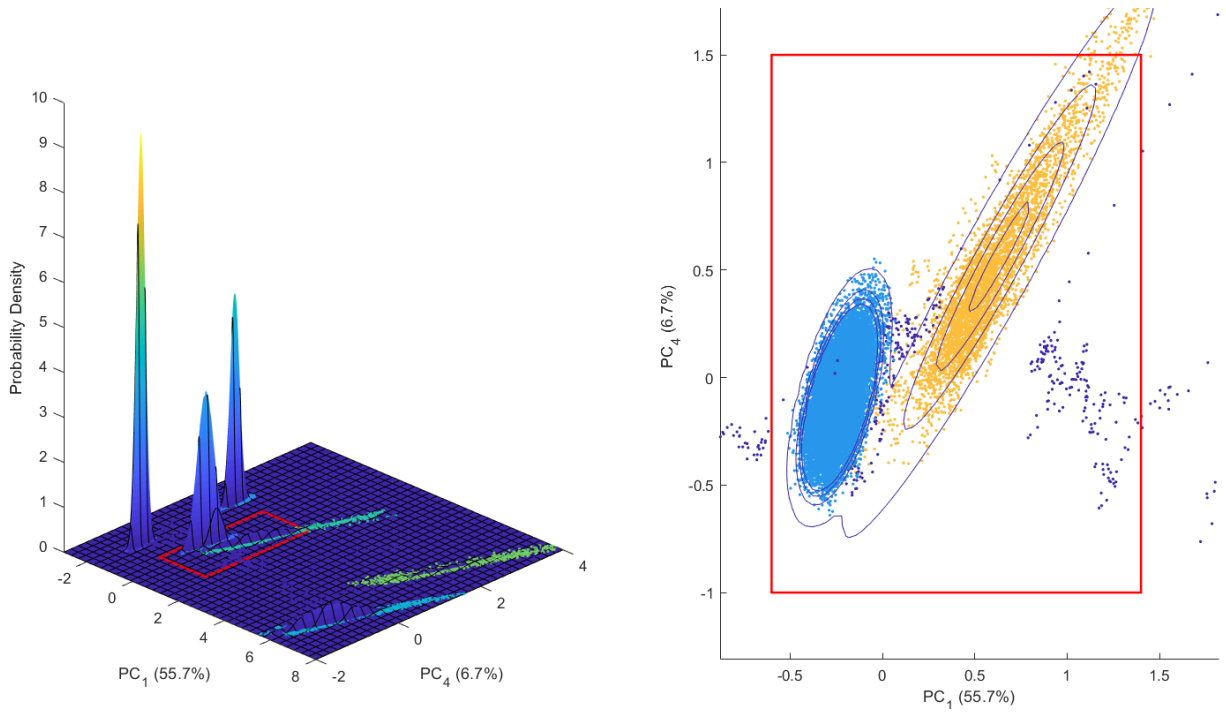


Figure 31: a) Surface Plot of the GMM fit on the CSTR data containing only stationary data b) Red bounding box in (a) magnified as a contour plot with level intervals [0.01:0.05:0.2]

The convergence per iteration can be seen in Figure 32. Strictly speaking a direct comparison of stationary and transient data convergence due to the nature of Equation 2-30 is not possible. Since all data samples are considered within Equation 2-30, in theory if a GMM is fit perfectly, a dataset containing both transient and stationary data should achieve a higher maximum likelihood. In this investigation however the transient data made up only a small percentage of the entire set, therefore it seems valid to compare the convergence of both cases.

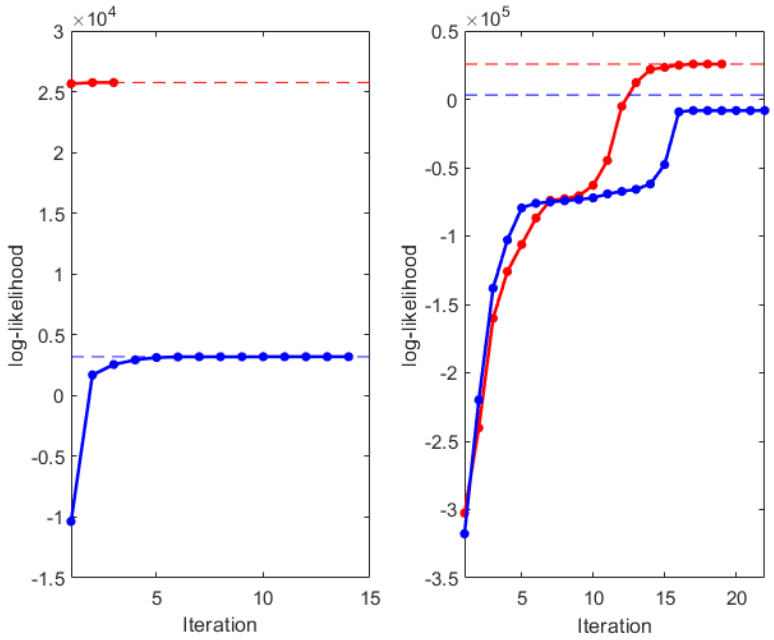


Figure 32: Comparison of EM convergence on only stationary data (red) and all data (blue) a) Initial Parameters obtained via K-means b) Default Parameter Estimation, where the dashed lines indicate the maximum log-likelihood achieved in (a)

Figure 32 a) shows the EM convergence when the EM initialisation parameters (Θ_0) are obtained from the K-means algorithm. Here it is clear that the EM algorithm converges faster (within less iterations) as well as to a higher log-likelihood when only stationary data is considered, even though less samples occur within the analysis (perhaps indicative of convergence to a local maximum likelihood). Further, it can be seen that the initial parameter estimates obtained from K-means (iteration 1 in Figure 32 a)) lie closer to the maximum likelihood, it can therefore be deduced that K-means clustering is more effective on only stationary data.

Alternatively only mode mean estimates are obtained via K-means. Mode/Gaussian weights are assumed to be uniform and the covariance is initialised to be equal to the variance of the dimensions. That is the covariance is set to a diagonal matrix (variables are independent). MATLABS default method of parameter initialization is achieved in this manner. Figure 32 b) displays the EM convergence with Θ_0 set in this manner. Comparing Figure 32 a) and b) it can clearly be seen that the initialization parameter estimates were not as close to "global" maximum. This is most likely due to the fact that the covariance estimate is poor, since describing a single mode covariance with the variance of a multimodal dataset will not be very effective. Even though it took more iterations to reach the assumed "global" maximum (dashed

lines) for the stationary data with a poor parameter initialisation, the same maximum log-likelihood is achieved (as in Figure 32 a)). This is not the case for the transient data, since it converges to a lower maximum than in Figure 32 a). It can therefore be deduced, that if the initialisation parameter estimates are not ideal, the EM algorithm is more likely to converge to a global maximum when only stationary data is analysed. It is thus assumed that the entire GMM fitting procedure becomes more robust to poor initialisation parameter estimates.

4.3 State Based Analysis Applied in a Unsupervised Approach

As discussed, the number of modes present within the data sets is usually not available. Further, the selection of specific PCs to represent data is also unrealistic. Investigations performed in 4.2.1.1 were done purely for visualisation and demonstration purposes. In this investigation however the unsupervised data analysis approach will be revisited, thus assuming that information seen in Table 8 is unknown. In this investigation the unsupervised analysis approach of the simulated CSTR data (described in Figure 11) with and without transients (removed via the SSD tuning described in Table 11) are compared.

As in section 4.2.1.1, PCs have to be selected to describe the data in a reduced dimensional space. Since the true structure of the data is unknown, the choice of PC selection will be purely based on the scree test described in 4.1.2.2. According to Figure 19, the first 4 PCs should be sufficient to describe the CSTR dataset, explaining 98 % of the variance.

4.3.1 Determination of the Number of Modes

Strictly speaking the analyses performed in 4.2.1.2 and 4.2.1.3 were biased towards the analyses of stationary data, since the dataset with transient data contained more than 6 states. Transient data could be described by Gaussians as well, however due to their most likely non-linearity and dynamic behaviour, a single transient would have to be described by multiple Gaussians (Zhang *et al.*, 2015). Therefore if more modes are considered when fitting the GMM, the stretching mode covariance may be reduced, and monitoring could improve. Gaussians containing transient data would however have to be distinguished from Gaussians containing quasi steady states.

Increasing the number of Gaussians when fitting a GMM however also increases the complexity of the fitting (EM) procedure. The EM procedure suffers from two major issues that have to be addressed. The issue is not only that the number of modes must be known prior to fitting but also the following (Figueiredo, Member and Jain, 2002):

- EM is highly dependent on the initialization parameters (especially the mean). Similarly to what is seen in Figure 32, if poor initialization parameters are utilized, the algorithm will converge to local maximum, poorly fitting the desired modes.
- The EM algorithm may converge to the boundaries of the parameter space. When the number of modes fit becomes large (larger than the true or optimal number), one of the mode weights may approach zero during the M-step. The corresponding covariance matrix may also not be positive definite (singular), non-invertible, and therefore Ill-conditioned.

As a result the EM algorithm may converge poorly or not at all. As mentioned in this investigation K-means clustering will be implemented to determine the initial parameters. Further, to prevent covariance matrices from becoming ill conditioned, a small regularization value (0.00001) will be added to the diagonals of the covariance matrices during EM. Therefore the covariance matrices are constrained to remaining linearly independent. This may result in a reduced maximum likelihood, however since the regularization value is so small, its effect is considered minor.

Figure 33 shows the Bayesian Information Criterion (BIC) obtained at different GMM configurations (as described in 3.2.3). Here the covariance matrices were not constrained (except for the regularisation value). Each EM was initialised by the centroids obtained from K-means. The initial estimates for the mode weights were uniform and all covariance were assumed to be a diagonal matrix of the considered dimension variances (MATLAB default).

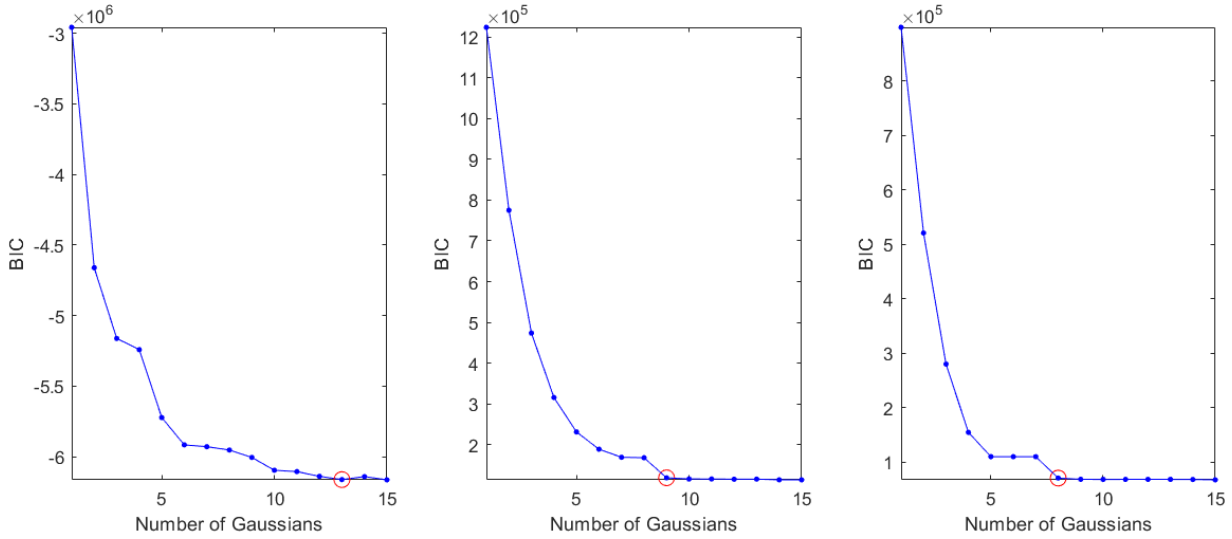


Figure 33: BIC for different GMM configurations, where the best determined setups are indicated by red circles. a) BIC determined from CSTR data including transitions on all standardized 14 process variables b) BIC determined from quasi steady and transients CSTR data on first four PCs c) BIC determined from only stationary CSTR data on first four PCs

Firstly, a GMM configuration containing 13 Gaussians was determined to best describe the standardised high dimensional space, as seen in Figure 33 a). On the other hand Figure 33 b) shows that a GMM with 9 Gaussians should best describe the transient and stationary data within the first four latent dimensions. That is that the likelihood is maximised without overfitting the data.

Secondly, for the stationary data and EM initial estimates obtained by K-means it can be seen that the best GMM configuration (according to BIC) consists of 8 Gaussians, as seen in Figure 33 c). This is surprising, since it is expected for the best GMM configuration to consist of 6 Gaussians, same as the true number of modes. Especially since all the transients were removed prior to GMM fitting. It can therefore be concluded, that the BIC suggests an overfitting model as a result of poor initialization estimates obtained from K-means. Since poor initialization estimates are provided, the EM converges to a local maximum when 6 Gaussians are considered. Figure 34 a) clearly shows that K-means clustering at this

configuration resulted in a poor data segmentation, since single clusters are described by various indices or multiple clusters described by a single index. These issues are then carried forward to the GMM fit, resulting in the EM converging to a local maximum. This therefore explains the high BIC seen for 6 Gaussians in Figure 33 c).

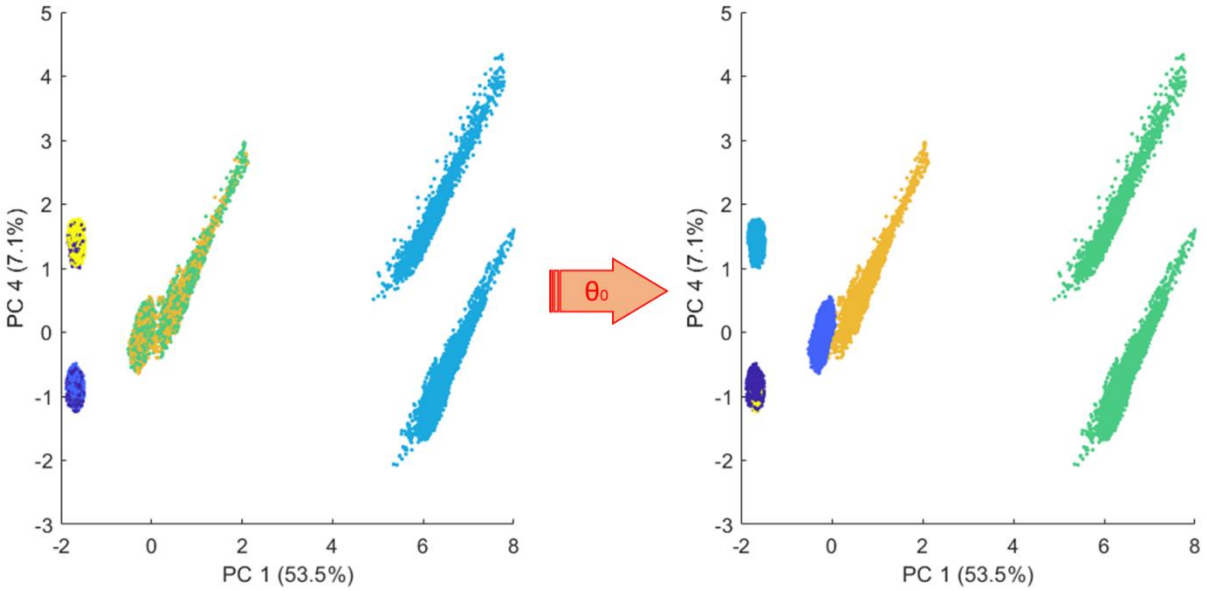


Figure 34: a) K-Means clustering result obtained when the first four PCs are considered, however displayed in PCs 1 and 4 b) Clustering result obtained from GMM using highest posterior probability again displayed in PCs 1 and 4 for visualization purposes

Firstly, K-means clustering may fail as the result of the curse of dimensionality. As mentioned in 2.3.2.1, when the dimensionality becomes high, the distinguishability of the Euclidean distance metric reduces. More dimensions increase the Euclidean distance, such that if infinite dimensions are considered, the Euclidean distance will also extend to infinity. Thus making it impossible to compare. Since the dimensionality here is however fairly low, the curse of dimensionality should not be too much of an issue.

Secondly, as seen in Figure 26, the inclusion of PCs 2 and 3 don't seem to introduce much mode distinguishing variance. These variables may therefore carry more inter-mode variance as opposed to intra-mode variance. In the context of segregating/clustering modes this is not ideal and can therefore be seen as variables introducing only noise. For example, linear discriminant analysis (LDA) which closely resembles PCA, is often used as a dimensionality reduction technique prior to classification. LDA finds discriminant vectors that result in the maximum ratio of between-class (inter-mode) variance to within-class (intra-mode) variance and are thus able to compute a reduced dimensional space that is ideal for classification (Choi, Park and Lee, 2004). However since the classes (modes) are not known prior to the analyses, LDA cannot be utilised. PCA is rather used to compute orthogonal vectors which result in the maximum variance (as described in 2.3.2). It should therefore be clear that the fundamental premise of PCA is not to segregate modes and as a result certain PCs may not improve clustering.

Lastly and most importantly in this case, the CSTR dataset is imbalanced (skewed distribution). Figure 35 b) clearly shows that the CSTR data is not perfectly centred on the multivariate mean. Since the variables were standardised prior to PCA, if the dataset was balanced, one would expect all modes to be equidistant from the multivariate mean. In this CSTR dataset (Table 8) however it is clear that only certain modes are near the multivariate mean, samples of these modes make up most of the dataset as seen when comparing Figure 35 a) and b).

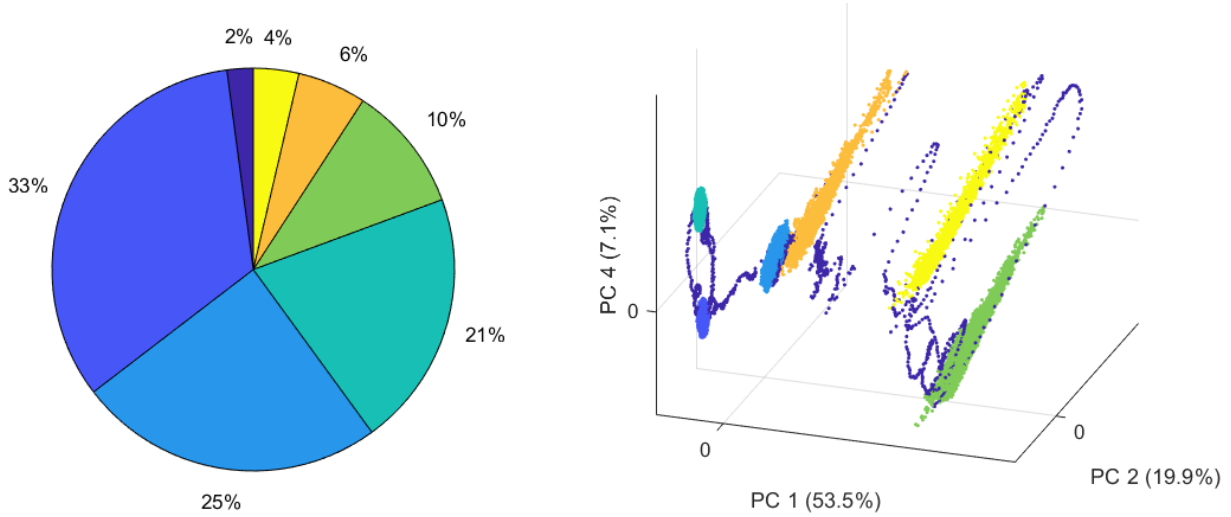


Figure 35: a) Pie chart describing the percentage of the total duration the CSTR was in each state b) Latent space separating the various modes, colours in both figures correspond

The issue of class imbalance is rarely discussed within the context of process monitoring or process optimisation, this should however not downplay the importance of the issue. As seen in Figure 34, class/state imbalance can have a detrimental effect on the analysis. The yellow and green modes are some of the "lighter" modes (seen in Figure 35), K-means clustering groups these modes into a single cluster, as seen in Figure 34 a). On the other hand, the darker blue modes (not transients) are the "heavier" modes, which K-means clustering splits into various clusters, as seen in Figure 34 a). This phenomena is described as the "uniform effect", where the majority class is divided into various minority classes, resulting in clusters having relatively uniform sizes (Xiong, Wu and Chen, 2009).

The issue of class imbalance does not only reduces the performance of K-means, but also diminishes the viability of performing PCA on a dataset. Since PCA only considers the mean and covariance of the data, it is a globality-based projection technique and therefore fails to extract local intrinsic information (Yu, 2016). For example, if a mode (quasi-steady state) occurs within process data, but only occurs for an extremely short duration relative to the other modes, the variance it contributes to the dataset is minor. Even if it is considerably different to the other data. PCA could deem this mode/variance direction as "irrelevant" and assign this variance direction to components which don't describe a lot of variance. Since the scree test only selects variables based on the cumulative variance explained, this mode may be "lost" from the data in the PC space.

Similarly, traditional PCA only minimises the sum of squared errors, if data however contains outliers, these may dominate the sum (Ding, 2006). Outliers are bound to occur within process data. For example,

operations which are completely different to normal operations, such as shutdown or start up procedures. Such data may dominate the sum and may alter the PC directions such that the latent space does not effectively describe normal operations.

In the case of this CSTR dataset, all modes are only distinguished once the 4th PC is considered. In other words, PCs 1, 2 and 3 carry more variance than PC 4, but don't effectively distinguish all modes. It is assumed that due to the data imbalance or the presence of outliers (start-up), PC 2 and 3 explain intra mode variance rather than important process behaviour (multimodality). Since PCs 2 and 3 explain more variance, they also contribute more to the Euclidean distance, however without distinguishing modes. This issue propagates to K-means clustering, since it only tries to minimize the sum of distances. As a result of these issues, K-means clustering fails, providing poor initial estimates to the EM algorithm of GMM which therefore converges to a local maximum likelihood.

These issues are difficult to overcome and may require a completely different approach. Methods of locality preserving projections (LPP) may assist with the issues of PCA and subsequently a more advanced clustering approach could improve on K-means clustering, such as DBSCAN (Thomas, Zhu and Romagnoli, 2018).

4.3.2 Evaluation of GMM for Mode Identification on Testing Data

In this investigation different GMM configurations are chosen to identify the various modes. Once again, here the effectiveness of determining a GMM from both transient and stationary data is compared. Model configurations resulting in the lowest BIC, as seen in Figure 33, were chosen as discussed in step 6 of the state based procedure (3.2.3). The training procedure until this far is therefore purely unsupervised, "expert knowledge" was however introduced in SSD tuning and setting of a regularisation value. It should be noted that testing a mode identifying model (monitoring) is only possible if the ground truth of the process is known or in other words the samples are labelled. In this case, the process was labelled based on the procedure described in section 3.1.2.

As seen in Figure 33, the best BIC occurs when the number of Gaussians exceeds the number of modes. This is an issue from the evaluation perspective, since the true identifying indices of the modes have to correspond to the identifying indices of the model. This is however not possible if the number of fit modes exceeds the true number. The only way a confusion matrix can therefore be obtained from the testing data, is if methods of mode merging/deletion occur prior to the evaluation, if possible. The GMM model therefore has to undergo a refining procedure with the help of expert knowledge. The refining procedure is discussed in steps 8 and 9 of the state based procedure (3.1.2).

An issue of GMM is said to result from the fact that when data is clustered, the sequence of the information is not taken into account (Song, Tan and Shi, 2016). Strictly speaking, this is true, since the convergence of the EM algorithm is purely based on the spatial information of data within the input space. However, subsequent analysis of the training data mode assignments can be performed, thus the time sequence of data can be accounted for and leveraged.

In this investigation expert knowledge of the dynamic behaviour of the CSTR is implemented to assist with cluster merging, such that the same number of true clusters occur within the GMM as in the ground truth. The approach is as such, the time sequence of modes within the data is analysed. If the determined modes switch between each other repetitively, they most likely describe the same mode. These modes can then be merged, and used as new initialisation parameters when the GMM is refit (this procedure is described in 3.2.3 step 8).

As seen in Figure 36 b), modes 4 and 6 as well as modes 2 and 3 switch to and from each other an unrealistic amount given the dynamic behaviour of the CSTR. It can therefore be concluded that these Gaussians most likely describe the same state. Thus the number of Gaussians within the GMM that describe the training set should be reduced to 6 and refit with the new initialization parameters based on the merged Gaussians. Figure 36 a) on the other hand does not display any unrealistic switching patterns, as a result it was decided not to merge any of these modes.

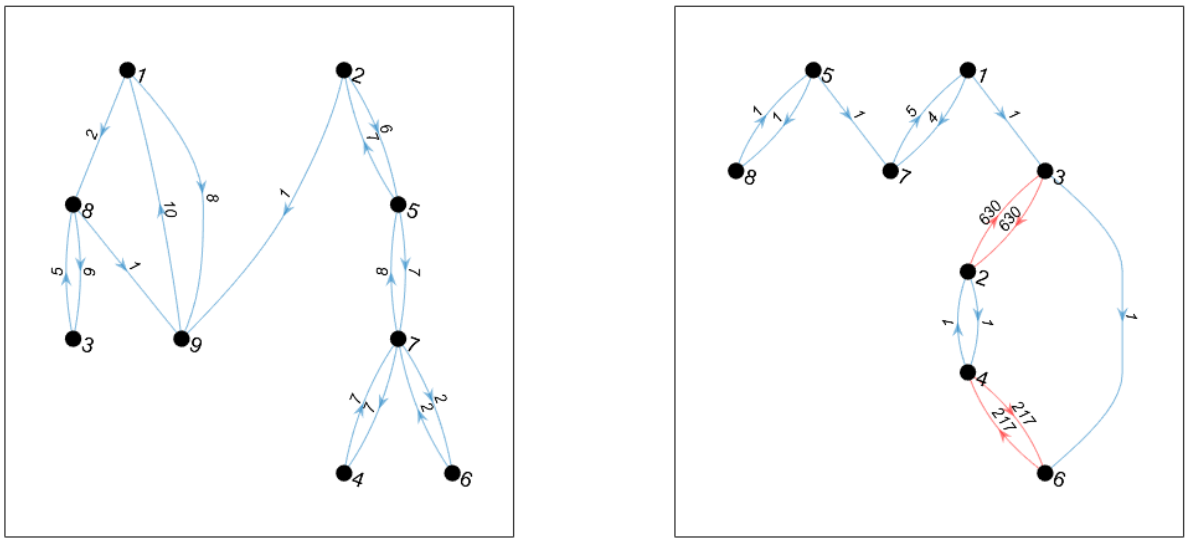


Figure 36: Connectivity graphs where nodes are the various Gaussians within the GMM fit to the training data and edge weights are the number of times these modes followed after each other sequentially a) Best BIC configuration for Steady and Transient Data (Figure 33 a)) b) Best BIC configuration for Steady Data only (Figure 33 b))

It should however be noticed that the number of remaining modes remain above the true number of modes for Figure 36 a). This occurs as result of the transient data present within the analysis, to which most likely three Gaussians are fit within the GMM. Therefore if stationarity analysis is not effectively performed prior to fitting of a GMM, an additional refining procedure (step 9 in procedure 3.2.3) to the GMM is required.

Comparing Figure 37 and Figure 35 a) data make up percentages, it can be seen that determined data distribution reflects the ground truth quite well. The issue however is that the remaining approximately 2 % is split into various states, these states are however transient and should not be considered for mode identification.

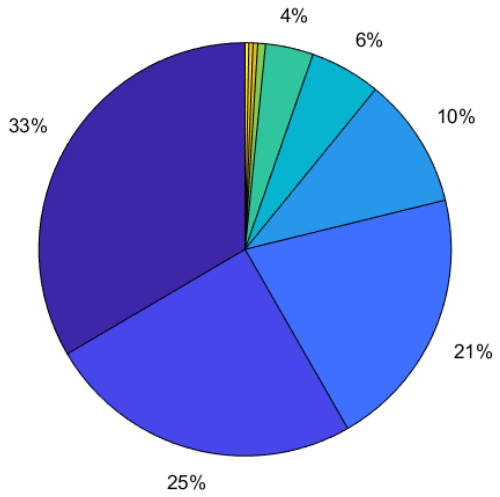


Figure 37: Pie chart describing percentage of various states in the CSTR data set obtained from the highest posterior probability of the GMM fit in Figure 36 a) (data including transients)

Here a minimum GMM Gaussian weight (0.037) is set, such that any Gaussians below this threshold are removed from the GMM along with the data that is assigned to these Gaussians (based on highest posterior probability). It is crucial for the relevant data to be removed as well, so that it does not have the stretching effect described in 4.2.1.3 on the remaining Gaussians (when the thresholds are determined). The process of removing such Gaussians from a GMM is denoted as filtering (described in 3.2.3 step 9).

The performance of the state identification procedure is shown Table 14, which was determined as discussed in 3.3.2. Here, the various variations of the developed state analysis procedure (3.2.3) are performed on the training data and evaluated on testing data (obtained as discussed in 3.1.1). The best BIC configurations here refer to those indicated in Figure 33. The supervised configurations refer to those discussed in 4.2. Further Table 14 indicates which steps of procedure 3.2.3 were performed, resulting in the performance. PCA here refers to if the dimensionality of the data was reduced, stationarity analysis refers to if the detected transients were removed from the training set (at SSD tuning described in Table 11), K-means refers to if the clustering algorithm was used to determine initial parameter estimates. Merging refers to if Gaussians were merged and finally G-filtering refers to if the removal of Gaussians and their assigned data occurred within the procedure. Further, a transient detection delay was also implemented, such that short transient periods (shorter than three sequential samples) are set to the most frequently occurring mode within the window. Detection delays can be used to suppress false alarm rates, in this case false transients. Details on detection delays are discussed by Addo (2019).

Table 14: Evaluation of various CSTR process state identification procedures on testing data with an 18 minute delay

Best BIC Configuration	PCA	Stationarity Analysis	K-means	Merging	G-Filtering	Precision	Recall	F1-score
a) Best BIC on standardized variables	×	×	✓	✓	✓	0.919	0.932	0.924
b) Best BIC all data	✓ 4 PCs	×	✓	×	✓	0.954	0.944	**0.948**
c) Best BIC steady data only	✓ 4 PCs	✓	✓	✓	×	0.950	0.946	0.947
Supervised Configurations								
d) 6 modes all data	✓ 4 PCs	×	✓	×	×	0.740	0.726	--
e) 6 modes steady data only	✓ 4 PCs	✓	✓	×	×	0.776	0.778	--
f) 6 modes all data	✓ PCs 1 & 4	×	✓	×	×	0.873	0.868	0.858
g) 6 modes steady data only	✓ PCs 1 & 4	✓	✓	×	×	0.927	0.922	0.924

As seen in Table 14 the various best BIC configurations considered in Figure 33 are evaluated, as well as additional configurations that may be applicable. These results however only display the macro averages of the evaluation metrics, the separate evaluation metrics for each state can be seen Appendix A Table 21.

As described in section 3.1.2, the GMM training (obtaining thresholds) and testing procedures that make use of NLLP. If the NLLP of a sample lies above the chosen local threshold (based on the Gaussian with the highest posterior probability), the state of the CSTR is deemed to be transient. If however the NLLP lies below the local threshold, the process is deemed to be within that mode. This procedure can be seen in Figure 38, where the NLLP lies above the thresholds for most of the transition. From Table 20 in Appendix A it can be seen that the transition resulted due to a set point change in the level controller. Further it should be noticed that the NLLP threshold varies based on a samples highest posterior probability, thus the threshold is a local threshold.

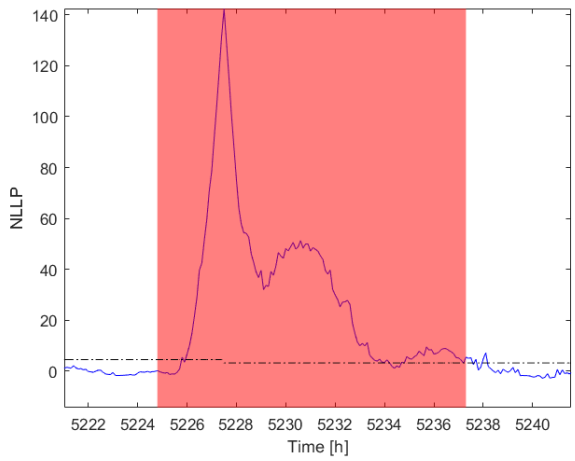


Figure 38: NLLP determined on testing data with training configuration c) in Table 14, where the red shaded region denotes a transient

Figure 38 however also shows that the NLLP remains below the threshold for a short duration while the transition has occurs. Similarly, the NLLP exceeds the threshold even after the transition has settled. Both these effects effect the evaluation metrics displayed in Table 14. From Table 15 it can be seen that CSTR modes or quasi steady states are never mistaken for each other (at that configuration), however transition states are occasionally mistaken to be modes and vice versa. It is however clear from Table 14 and Table 15 that the developed approach to CSTR state identification works well on testing data.

Table 15: Multiclass Confusion Matrix obtained from testing data using state identification configuration c) in Table 14, where state 0 indicates a transition state

		Estimated State Label						
		1	2	3	4	5	6	0
True State Label	1	28724	0	0	0	0	0	59
	2	0	19423	0	0	0	0	35
	3	0	0	14263	0	0	0	80
	4	0	0	0	7710	0	0	11
	5	0	0	0	0	5841	0	86
	6	0	0	0	0	0	2364	11
	0	55	46	42	102	14	56	1079

From Table 14 it can be seen that both procedures b) and c) achieved good performance. Both these procedures made use of PCA, K-means clustering and GMM/data refinement procedures. However it can be seen that the state identification performance achieved without the removal transient periods prior to analysis (Table 14 b)) is even better than if SSD is implemented (based on macro F_1 scores). Thus, if analysis procedures of PCA, GMM, BIC, Gaussian merging and filtering are implemented, very much similar results can be achieved as with utilisation of SSD, with much less effort. The reason for the minor improvement in performance is assumed to be as a result of a larger training set. When SSD is performed, in this case all transients are removed, however false transients are also removed (FTR of 24 %). This results in less samples being available during GMM fitting and thus results in a poorer fit.

From Table 14 it also clear that if a specific number of modes (from expert knowledge) are fit to the data, the results are likely to be poor. This is the case for Table 14 d) and e), achieving worse performance based on recall and precision. Here the macro F_1 scores could not be calculated since modes were completed “lost” from the analysis. In both cases mode 6 was incorrectly identified throughout the testing set. This can be seen in Table 21 Appendix A, most likely due to reasons discussed in 4.3.1. It is therefore clear that the BIC (or any similar metric penalizing model fit quality with complexity) is crucial to effectively fit GMMs to data. Since the number of Gaussians required based on BIC usually is more than the true number of Gaussians, it is clear that a “quantity over quality” approach is required when fitting GMMs, however with subsequent GMM refining (merging/filtering). The improvement in performance compared to if the true number of Gaussians are fit to the data can be attributed to the fact that the EM algorithm is less likely to converge to a local maximum (more resilient to outliers and less

sensitive to varying degrees of variance in PCs). Thus it is better to fit more Gaussians and refine the GMM than to fit too few and converge to a local maximum.

It also has to be said that if steady state analysis can effectively be achieved on a dataset, it may be more beneficial than the filtering approach. Firstly, the chances of a mode being merged with a transient do not exist. The effect of transient data being considered can be seen in Table 14 f), here it is clear that the performance of a state identifying model diminishes, due to the stretching of the covariance matrices (compared to Table 14 g)). Secondly, the simplistic filtering procedure is not required and thus may improve the discoverability of "light" (few samples) modes, which may in fact be the interesting modes within a dataset. Data may be such that the weight of transient data is more than the weight of certain modes, thus filtering based on purely Gaussian weight could be problematic. In a sense data is blindly removed. This is an issue that becomes more prevalent with increased transient data in a set. Since data filtering using SSD takes into account more complex features of data, such as the sequence of data, it is a more effective filtering approach. Lastly, SSD may provide a set of estimates which experts can use to initialize the EM of the GMM. Filtering by means of SSD is however also more difficult to achieve, as discussed in 4.1.2.6.

Additionally the effect of the chosen variables is also investigated, as seen in Table 14. Comparing the performance of Table 14 b) and f), it can be seen that if 4 PCs are considered, the state identifying scheme is better. Thus it can be concluded that even if all modes are visible within a reduced dimensional space, further variables may still have to be included. Similar results can be seen when comparing Table 14 c) and g). Interestingly it seems that the state identifying (monitoring) approach works quite well even if all CSTR variables are considered as seen in Table 14 a). At higher dimensionality however a detection delay becomes more important, due to reasons discussed in 2.3.2.1. It assumed that the decrease in performance relative to Table 14 b) for example is as a result of the GMM overfitting to the training data. The number of parameters required to be estimated by the EM algorithm was 1266 (after merging), this far exceeds the number of parameters required when only the PCs are considered. As a result the GMM is extremely sensitive to slight variance from the training dataset to the testing dataset. This can clearly be seen by the low precision of the transient state (state 0) in Table 21 Appendix A. Samples of various modes are continuously mistaken as transients, similar to increased false alarm rates. The decrease in performance is however minor thus it seems that the procedure is able to effectively deal with issues of high dimensionality. This is important to note, since PCA may not be useful in all situations. Especially when data contains a large number of outliers, which may dominate the PCs.

Based on the results in Table 14, the described monitoring approaches work quite well. It is clear that a good state identifying approach makes use of GMM, BIC, the sequence of data (merging) and some sort of a filtering technique (SSD or based on minimum weight).

4.3.3 Mapping process states

Now that the applicability of the state identifying approach has been proven to work quite well on testing data (online) for both methods using SSD and without, the training data can be mapped as described in 3.2.4. It should therefore be clear that the process map is determined from the training (historical) data, however the map would be of no use if the online state within the map is not identifiable. The monitoring (state identifying) approach was thus evaluated prior to mapping the states, in industry this would not be the case.

SSD is still implemented within the connectivity analysis (3.2.4), due to the fact that it provides an ideal data structure on which targeted transition analysis can be performed. Adjustments of the procedure discussed in 3.2.4 can most likely be made such that SSD is not required, however since SSD was found to work quite effectively on the simulated CSTR data set, development of this procedure is not considered within this investigation.

As discussed in 2.1.2.2, various factors need to be considered when optimising processes. The optimisation could be achieved with the use of a fundamental model, but these models are sometimes difficult to develop and may not account for abnormal process behaviour. Human supervisory control therefore remains critical. Thus as mentioned in 1.3, key features of the data-driven approach should be to able identify the current state a process is in (investigated in 4.3.2), identify operating states that maximise a certain economic objective (KPI) and determine the set of actions required to reach these states, if possible.

However since process states are usually only discernible within higher variable dimensions (Aldrich *et al.*, 2014), efforts should be made to reduce the dimensionality of the state space such that it is interpretable by human supervisory staff. Connectivity graphs as seen in Figure 36 are ideal techniques that digest states present at high dimensions into a 2D state map, which could effectively describes the relationship between states. The 2D state map with the various KPIs for the states obtained from procedures discussed in 3.2.4 for the CSTR dataset described in 4.1.1 can be seen in Figure 39.

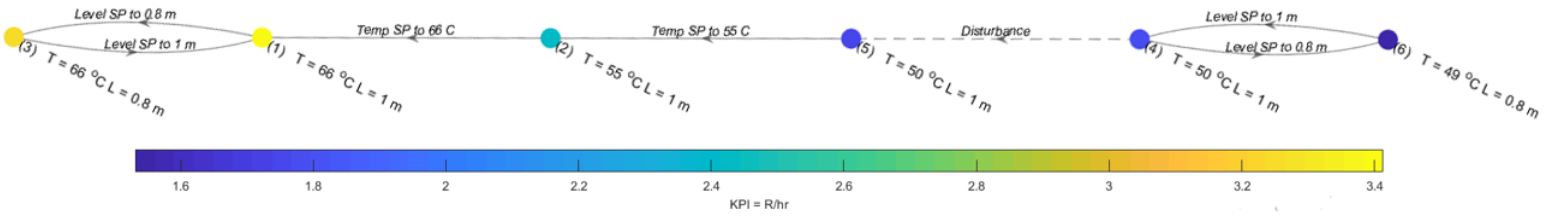


Figure 39: CSTR Data Map obtained using the procedure described in 3.2.4

Data seen in Figure 11 and Table 20 Appendix A is therefore condensed into an easily interpretable map, which can be effectively used by human supervisory staff to operate the CSTR. Here the node edges describe the transitions between the modes of operation (the nodes), thus displaying the accessibility of the various modes. It should be noticed that the node edge description displays the procedures required to shift from one process state to another. The node description in this case described the average

controlled variable conditions for the various modes, which are the CSTR temperature and the CSTR level. The node/mode colours are set to the KPI of the mode (as determined in 3.2.4), thus CSTR can be operated according to its economic objectives using the map. All in all such a data map will assist human supervisory staff to effectively navigate processes.

4.4 Unsupervised Approach Evaluated on a Complex CSTR Dataset

To demonstrate the usefulness of the state map and the ability of providing actionable advisories, the developed state based system is evaluated on a more complex CSTR dataset. This dataset contained 15 unique modes and extends a period of 9000 hours. For the generation of this dataset however the minimum steady state time is set to be shorter than in the previous investigations, thus a larger fraction of the dataset contains transient data (see section 3.1.1). Approximately 7 % of the dataset is transient, more details about the training and testing data set can be seen in Appendix B. The analyses here will be performed as in 4.3. Thus procedures 3.2.1, 3.2.2, 3.2.3 and 3.2.4 are all performed on the training dataset seen in Appendix B Figure 55 a). Based on the scree test, 5 PCs are considered within the analysis, explaining 99 % of the variance, as seen in Appendix B Figure 55 b).

As with the analysis in 4.3, the state analyses (3.2.3) is evaluated for both analyses with and without SSD. The BIC achieved when all transients are removed prior to the analysis and not removed can be seen in Figure 40. Transients are removed using the SSD tuning described in Table 24 in Appendix B, achieving the visual results displayed in Figure 56 (Appendix B).

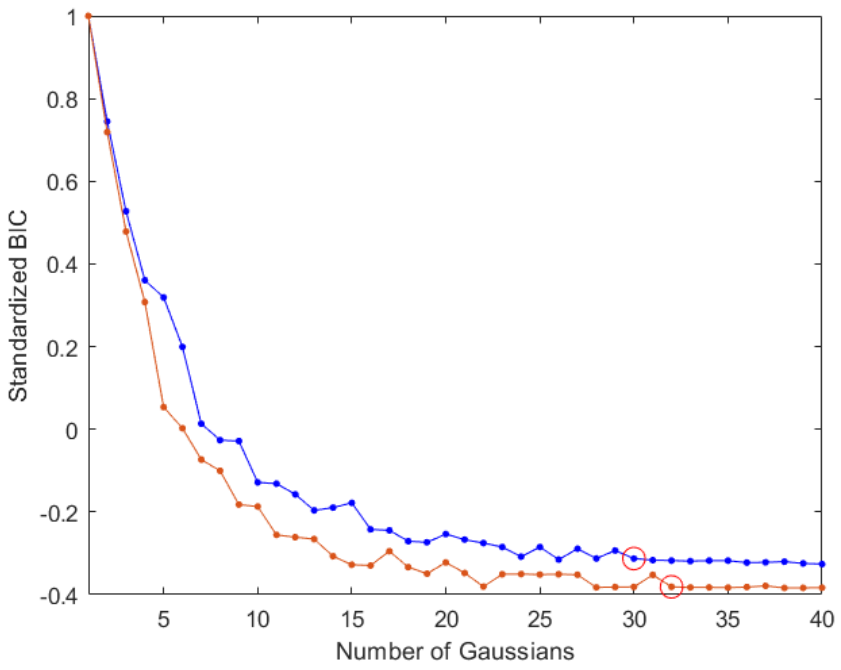


Figure 40: Standardized BIC for Complex Dataset described in Table 22 where detected transients are removed prior to the analysis (orange) and containing transients (blue)

It should be noted that the BICs in Figure 40 were standardized by dividing each set by their maximum, this was done for visualization purposes only. Based on Figure 40, a GMM configuration containing 32 Gaussians would be best if transient data was removed prior to the analysis. Similarly, a GMM configuration containing 30 Gaussians would be best for the entire dataset.

4.4.1.1 Analysis Containing Transient Data

The information required to refine (steps 8 and 9 of procedure 3.2.3) the "transient containing" GMM can be seen in Figure 41. Usually, as in the previous analysis, Gaussian merging is performed prior to Gaussian filtering. The order at which these refining procedures should be performed should be determined by the expert performing the analysis. It is crucial to note that different output GMMs could be obtained based on the order of the procedure. Figure 41 shows the mode connectivity graph, here however the Euclidean distances between the mode means (only the connected modes) are used as attractive forces between nodes, such that modes with small distances between them lie close to each other in the graph. Further, the colour of the nodes indicate the Gaussian weight of each "mode".

Based on Figure 41, it is clear that certain modes should be merged, however for some cases, the decision to merge modes is less obvious. For example, modes 1 and 13 should be merged based on the large number of sequential switches and the relatively small Euclidean distance between these modes. Similarly modes 6/22, 10/25 and 11/12 should all be merged. A more clear view of the connectivity graph can be seen in Appendix B Figure 58. Modes 4 and 20, and modes 7 and 17 could also be merged. Based on the larger (relative) Euclidean distance between the modes it is decided not to merge these modes. Thus both the Euclidean distance and the number of sequential switches were considered during merging.

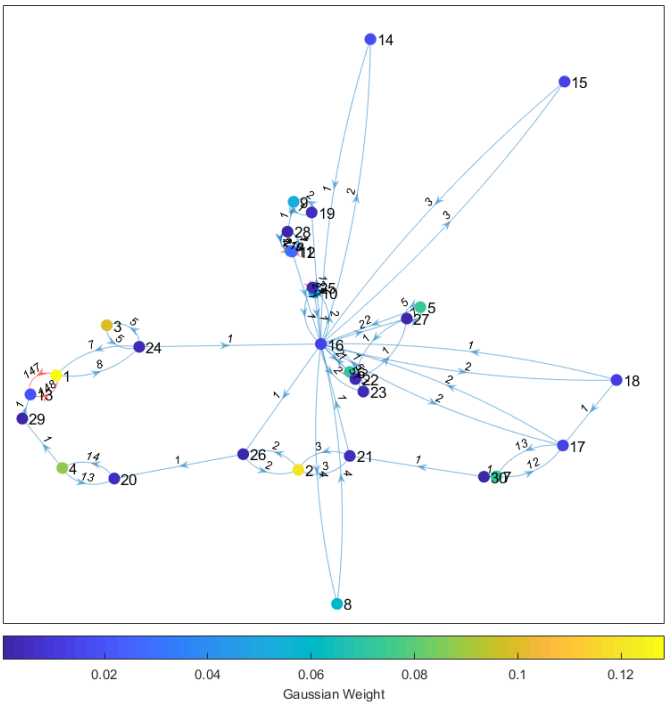


Figure 41: Graph assisting the GMM refining procedure based on the data sequence, Euclidean distance between mode means and Gaussian weights (node colour). Red edges indicate merged modes

Although the Gaussian weight is informative of perhaps the importance of the mode, it gives no real indication as if the Gaussian is transient or not. It is once again unclear which Gaussians to remove. This highlights a serious issue of suggested refining procedure. Removing Gaussians purely based on their weight may result in the loss of valuable mode information. The hope is that transient data is contained within the Gaussians of lower weight, however the larger the number of modes present within the data, most likely the larger the number of transitions within the data.

To allow a confusion matrix to be setup from the GMM, it is decided to set the threshold Gaussian weight to 0.012 (this is after merging), thus removing 11 Gaussians from the GMM and their assigned data (4 % of the dataset). Approximately 2 weeks of simulated data are removed. The remainder of the procedure 3.2.3 is performed and the state identifying performance of the resulting GMM on testing data (procedure 3.2.5 steps 1 to 4) can be seen in Table 16.

Table 16: Performance Evaluated as discussed in 3.3.2, here state 0 refers to the transient state

State	F1	Precision	Recall
1	0.997	0.997	0.998
2	0.986	0.989	0.984
3	0.995	0.992	0.998
4	0.994	0.992	0.996
5	0.989	0.993	0.985
6	0.997	0.998	0.996
7	0.991	0.986	0.996
8	0.994	0.990	0.999
9	0.991	0.992	0.990
10	0.994	0.995	0.993
11	0.985	0.975	0.995
12	0.949	0.903	1.000
13	0.981	0.965	0.997
14	0.404	0.253	1.000
15	0.773	0.642	0.971
0	0.188	0.605	0.111
Macro Mean	0.888	0.892	0.938

It should be noted that the determined identifying indices (see procedure 3.2.3 step 7) of states 14 and 13 had to be swapped. This is due to the fact that the true fraction of the data of states 13 and 14 were almost identical, making procedure 3.1.2 prone to error. Comparing macro results from Table 16 to Table 14, it is clear that the procedure 3.2.3 did not achieve as good of a performance. The transition states (state 0) and state 14 (mode 14) were identified poorly (based on F_1), however all other states achieved good performance. The low precision of state 14 indicates a poor Gaussian fit. The high recall yet low precision is a clear indication of stretched covariance, suggesting that the mode is usually detected when it occurs but many false detections occur as well. Specifically, transient data is often mistakenly identified

as mode 14 as seen in the confusion matrix in Appendix B Table 25. Transient data remaining within the analyses during EM convergence once again results in a stretched covariance (converging to local maximum).

A very important metric here is the F_1 score of the transients (state 0), which indicates the models ability in discerning a mode (quasi steady state) from a transient. It is clear that the model did not perform very well in this aspect. The low recall of state 0 again indicates that not all transient data was effectively filtered from the dataset. In general the model seemed to be able to identify true "heavy" modes quite well (since modes are indexed in descending order of Gaussian weight), but had more difficulty in discerning "light" modes from transients.

4.4.1.2 Analysis Containing only Stationary Data

The following analyses is performed as in 4.4.1.1, here however SSD is utilised to remove transients from the data. The ROC curve (determined using tuning parameters shown in Table 24 in Appendix B) can be seen in Figure 57 in Appendix B, achieving an AUC of 0.96. It is therefore clear that SSD works well for the specific dataset. When the stationarity threshold (Θ_{ss}) is set to 0.64, a FTR of 4 % and a MTR of 13 % is achieved, visual results of which can be seen in Figure 56 in Appendix B. Figure 42 once again shows the switching frequency, Gaussian weight and the relative Euclidean distance of the mode means, however with most transients removed prior to the analysis.

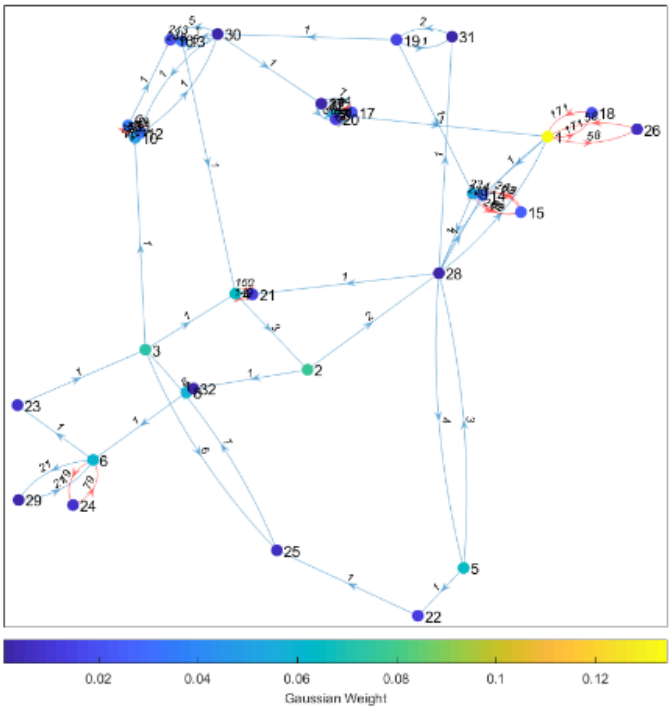


Figure 42: a) Considerations when refining the GMM based on the data sequence b) Considerations when refining the GMM based on Gaussian weight (after merging)

Here Gaussian merging (based on Euclidean distance and switching frequency) occurs prior to Gaussian filtering. This is also the suggested procedure, since Gaussian merging is actually based on fundamental concepts of process control. That is that a process should be operated smoothly (2.1), if excessive mode

switching occurs this is not the case. A more clear display of which modes were merged can be seen in Appendix B Figure 59. It is clear that modes that lie close to each other, usually have a high frequency of sequential switches. The overall decision to merge process is thus as follows:

1. If modes have large Gaussian weights (signifies their importance), large number of sequential switches and are close to each other in terms of Euclidean distance, they should be merged.
2. If modes have a large number of sequential switches, but lie far apart, then one should be cautious when deciding to merge the modes.
3. If either of the modes in 2. has a low Gaussian weight (assumed to be transient), then they should probably not be merged.
4. If modes have a small number of sequential switches, but lie close to each other, then the modes should probably not be merged.

Even though SSD was performed on the dataset prior to the analyses, SSD did not achieve a MTR of 0 % (rather 13 %), thus some transient data is still present within the analyses. Setting the filtering threshold to 0.006, 6 Gaussians and their assigned data are removed from the model (as discussed in 3.2.3). Issues as discussed in 4.4.1.1 apply here as well. The rest of the procedure 3.2.3 is performed and evaluated on a testing set (as described in 3.2.5 steps 1 to 4), the state identifying performance of the resulting GMM can be seen in Table 17.

Table 17: Performance Evaluated as discussed in 3.3.2, here state 0 refers to the transient state

State	F1	Precision	Recall
1	0.997	0.997	0.998
2	0.984	0.991	0.976
3	0.995	0.992	0.998
4	0.992	0.992	0.993
5	0.989	0.992	0.986
6	0.997	0.998	0.996
7	0.992	0.986	0.997
8	0.994	0.989	0.998
9	0.991	0.992	0.990
10	0.993	0.996	0.990
11	0.984	0.975	0.994
12	0.931	0.993	0.877
13	0.982	0.996	0.968
14	0.943	0.925	0.962
15	0.877	0.796	0.975
0	0.819	0.816	0.823
Macro Mean	0.966	0.964	0.970

Comparing Table 17 and Table 16 it is clear that performing SSD prior to the analysis allows the state based procedure to be more effective, reflecting the results seen in Table 14. The difference in performance here is however more profound. The major difference in performance is assumed to originate from the increased presence of transient data within the training dataset (refer to phenomena

discussed in 4.2.1.3). The less transient data present within the analysis, the better K-means clustering is able to provide good estimates of the initialisation parameters.

Table 17 shows that as in Table 16, the state identifying model seems to improve with increasing Gaussian weight. Modes containing less data will be more prone to the stretching effect of transient data. This can be clearly seen analysing the precision of state 15 ("lightest" mode). The lower precision on 0.796 indicates that mode 15 was falsely detected more often, a clear indication of a poor (stretched) covariance fit (ie. low precision, high recall). Further, it is clear that this GMM is far better at distinguishing modes from transients, as seen from the relatively high F_1 score of Table 17. More information of the state identifying performance can be seen in Appendix B Table 26.

4.4.2 Mapping the Complex Dataset Process States and providing Actionable Advisories

Here procedure 3.2.4 is utilised to map the process states, using the GMM obtained from 4.4.1.2. The procedure discussed in 3.2.4 could be adjusted such that SSD is not required (using a moving window for example). However, since it was determined that the use of SSD performs well on the discussed simulated datasets as well as improves the GMM, the development of this algorithm is not considered within this investigation.

The process map of the dataset can be seen in Figure 43. The process map seen in Figure 43 is clearly more complex than the map seen Figure 39, in this case providing an accurate mode map will be far more useful. Training data seen in Figure 55 a) Appendix B are thus digested into a 2D format, allowing humans to effectively navigate the considered process.

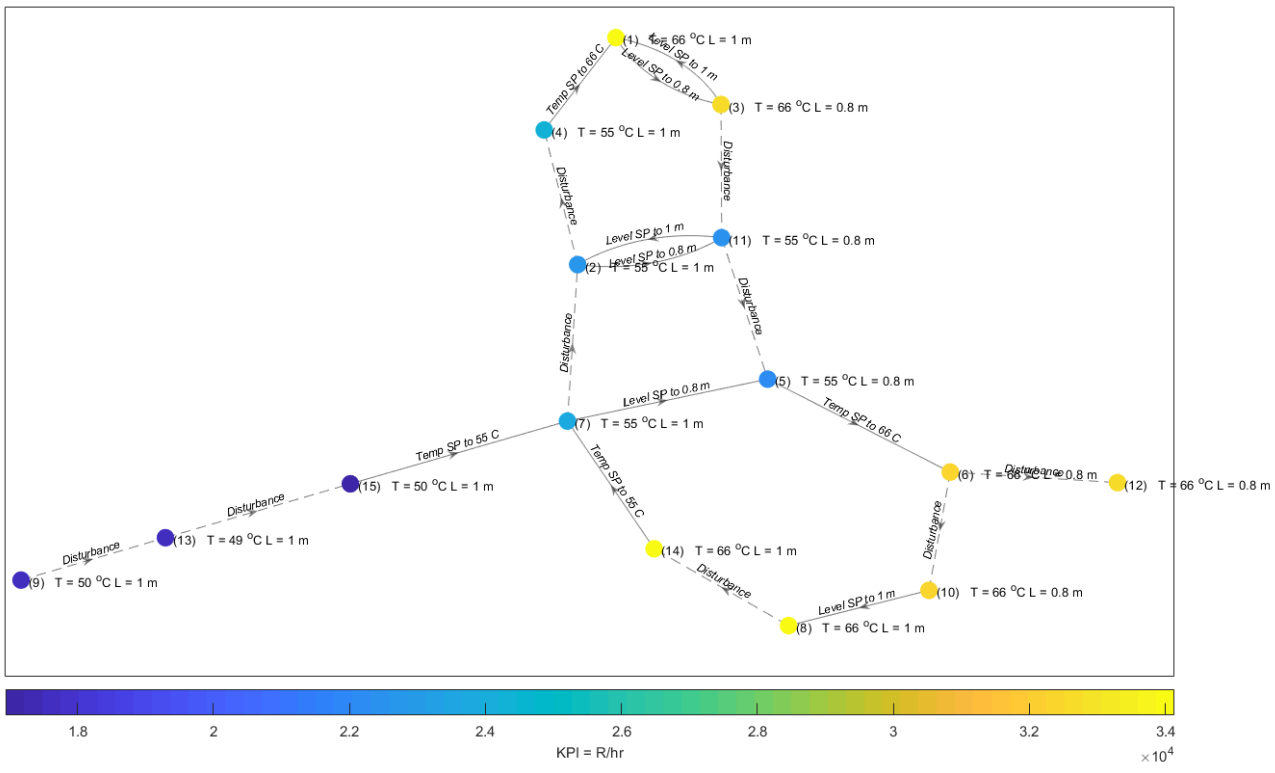


Figure 43: Process Map obtained using procedure 3.2.4 and the GMM obtained during investigation 4.4.1.2

From Figure 43 it can be seen that certain mode mappings are however not entirely correct. For example, the cause for a state shift from mode 3 to 11 is determined to be the result of a disturbance. A mode switch from 3 to 11 however never occurred in historical data (training data), rather a mode switch from 3 to 12 to 11. The reason for this incorrect mapping is determined to be due to the SSD, a true transient occurring within the training data was not detected. As a result procedure 3.2.4 failed to establish a connection between modes 3 and 12. Further, the connectivity (reverse) between modes 5 and 6 does not appear as well. It is therefore clear that supervisory staff needs to ensure that the produced state map reflects their experience. The remainder of the map is however an accurate representation of the ground truth map, which can be seen in Appendix B Figure 60.

A process map can be used in real-time operation, providing actionable advisories that can be used by supervisory staff to allow them to interpret, plan and finally execute operation more effectively. It should be clear that in Figure 44 a distinction between standard operating procedure (SOP) and disturbance caused transitions are made, this distinction is crucial when providing actionable advisories. In literature regarding optimality assessment, the details of the connectivity between modes are not taken into account (Ying, Li and Yang, 2020). Advisories are simply based on modes to switch to that have a higher economic index or KPI, but fail to take into account the fact that these modes may not be reachable.

For example, when the CSTR is within state 15 (approximately at samples 1000 of testing data) it is clear that the system is not operating optimally (mode 15 KPI is low), but rather at an “global” optimality of approximately 0.5 (determined as discussed in procedure 3.2.5). Actionable advisories in this case are therefore that the CSTR could be operated more effectively if it is transitioned to mode 6. This transition can be achieved by first changing the temperature controller set point to 55 °C, then waiting for a steady state to be achieved (mode 7). After which the CSTR level set point should be set to 0.8 m and steady state should be achieved (mode 5). Finally the temperature set point of the CSTR should be set to 66 °C to reach the most optimal accessible mode (mode 6).

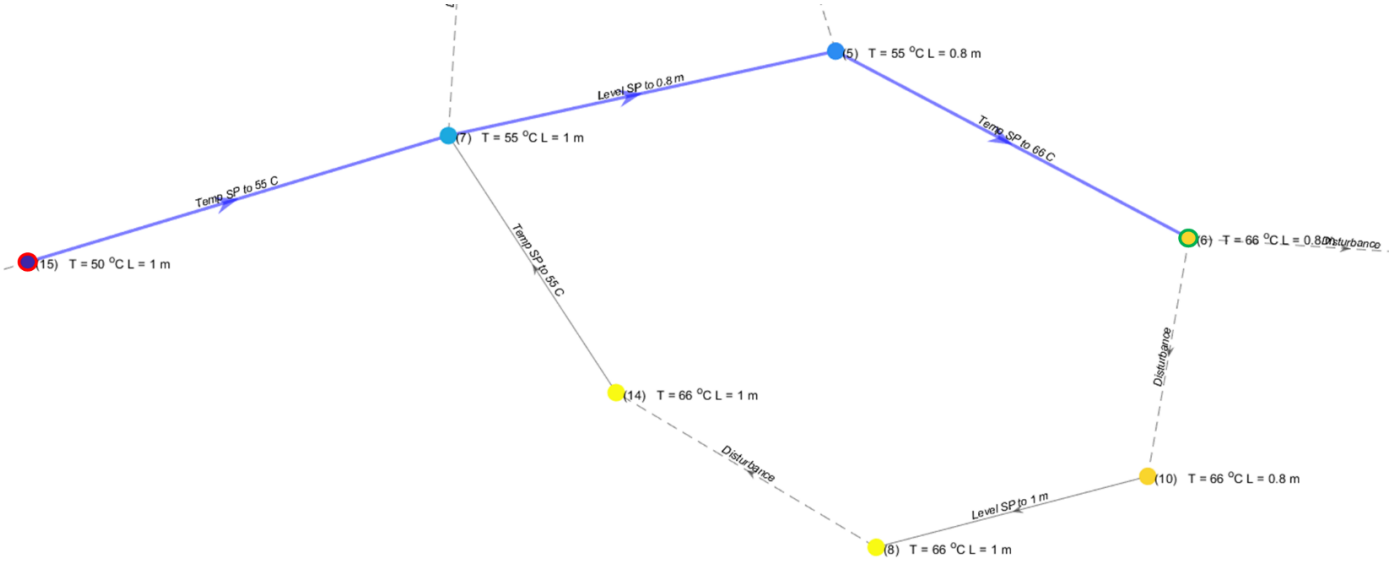


Figure 44: Example of Actionable Advisories Provided. Here the red node displays the current mode of operation, the green node displays the most optimal mode and the blue path indicates the advised SOP

What should be clear is that the actionable advisories are based on historic actions only. The system will never advise the human supervisor to perform an action that has never been performed in history (training data). This may increase the reliability of the state based decision support system, but will ultimately limit its versatility. The previous suggested procedure of shifting from mode 15 to mode 6 may not in fact have been the best advisory, since simply raising the CSTR temperature to 66 ^{o}C could have been better (shift to mode 14). However, since the data is generated randomly, this operation logic is not contained within the data, this should not be the case for industrial process data.

What is clear is that if the discussed data analysis procedures are performed perfectly (see 3.2) then the resulting state based system can provide actionable advisories in a “human-centred” manner, without the development of fundamental models. Thus raising the levels of trust within the control environment, as well as reducing the reliance of processes on their human supervisors. However as seen in the previous sections, perfectly developing such a model may be difficult even from simulation data. Actual process data may pose many more challenges.

5 APPLICATION OF DEVELOPED PROCEDURES ON INDUSTRIAL DATA

In this section the developed analysis procedures are applied to industrial process data obtained from an Anglo Platinum concentrator operation. This dataset contained over 8761 data samples collected over the period of a year (sampling rate of a sample per hour) for approximately 4100 variables. The site is made up of approximately 14 sections, each of which performs a different task such as crushing, milling, flotation, etc.

Based on a preliminary analysis of the entire dataset, it is decided that a state based analysis of the milling circuit of the plant could be useful. Specifically a single mill and its related variables are considered within the following investigation. A total of 62 variables and 8488 samples were considered within this analysis. A simplified process flow diagram of the circuit can be seen in Figure 45.

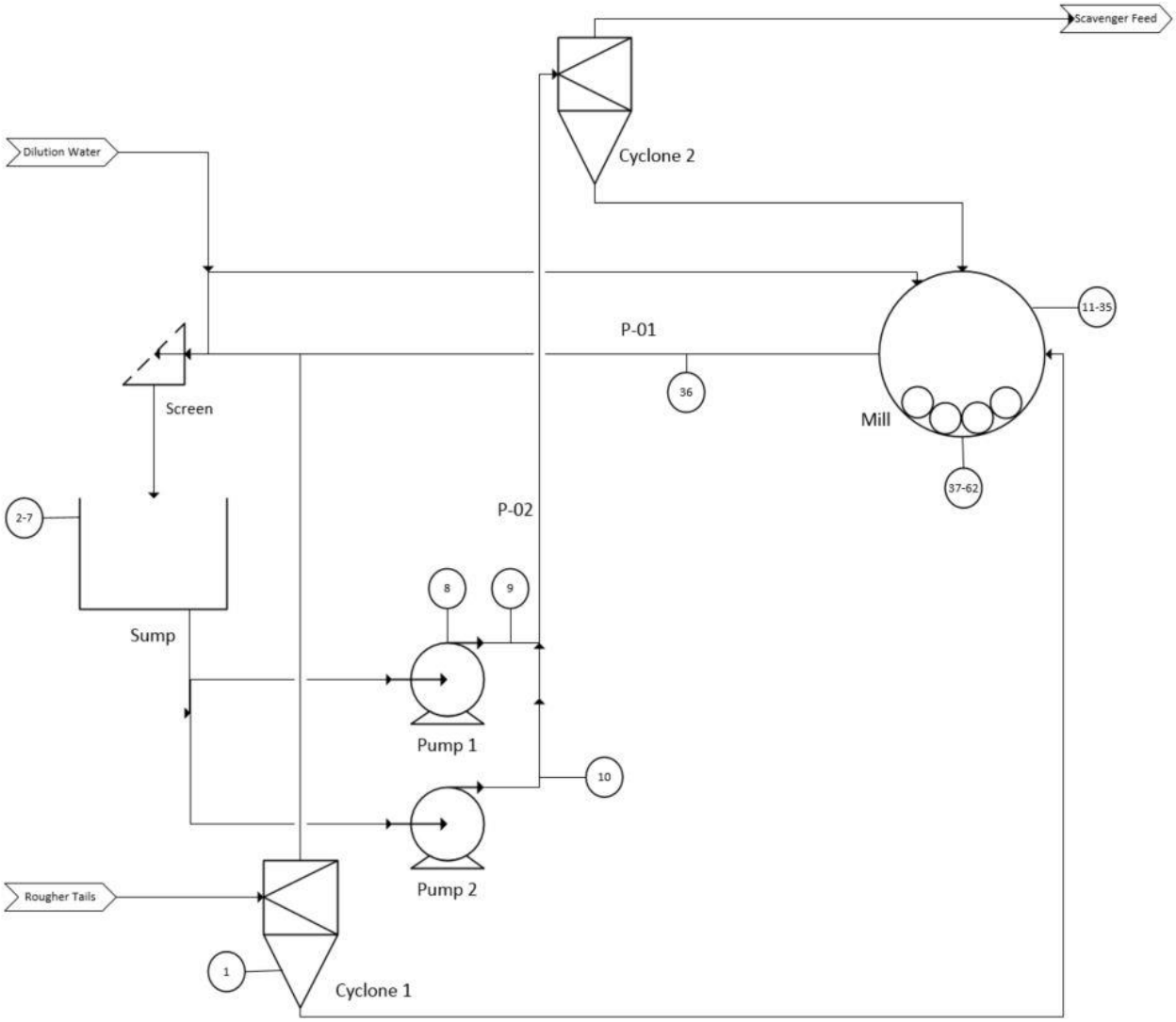


Figure 45: Simplified process flow diagram of the milling circuit. The sensor tag descriptions can be found in Table 27 Appendix C

5.1 Major issues of discussed approach on industrial process data

As discussed in section 2.2.1 industrial process data poses many challenges to the developed state based analysis which do not occur within the CSTR simulation data. All system support functions (see 2.2.2) were kept in mind during the development of the final state based model, except for the deviation analysis. Deviation analysis was not considered for the CSTR simulation data, since no process deviation (commonly known as process drift) was included in the model (see section 3.1). This is however not the case for industrial data and may pose a serious challenge to the developed approach. Addressing this system support function is however not considered within this investigation, further insight into these issues are discussed by A. Prince (2019) as well as Xie and Shi (2012).

Further, the connectivity procedure discussed in section 3.2.4 is considered to be too simplistic to be applied to this dataset. Although the procedure achieved a reasonably accurate mapping of both CSTR datasets, real process data is far more complex. The secondary milling section had a total of 7 controllers, each of which had variable set points. Two of these controller set points can be seen in Figure 46.

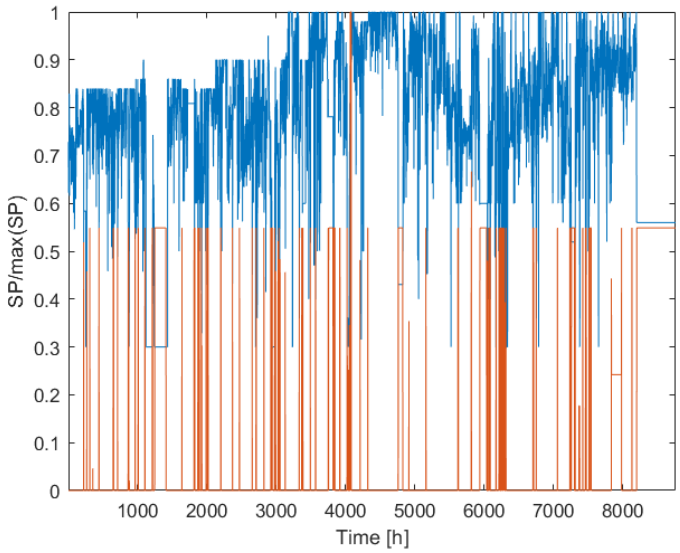


Figure 46: Blue shows the set points of the screen flow inlet controller and orange shows the set points of the pump flow

From the frequency of set point changes within Figure 46, especially that of the screen flow inlet controller, it is clear that some sort of adaptive control strategy is implemented within this process. Set point changes are constantly made throughout operation. Further complicating the analysis are hierarchical control layers such as regulatory, optimisation, and supervisory control. Thus distinction between set point caused transitions and disturbance related transitions will be difficult to determine, limiting the applicability of procedure 3.2.4. It is therefore clear that the approach described in procedure 3.2.4 is an over simplification of the actual approach that would be required, it does however demonstrate a valuable concept. The following industrial data analysis will thus only be concerned with procedures 3.2.1, 3.2.2 and 3.2.3.

5.2 Stationarity Analysis on Industrial Data

A total of 62 variables were considered within the state based analysis, these variables were selected based on an estimated variable quality. A description of these variables can be found in Appendix C. It is clear that the number of variables here high relative to the number of variables considered within the CSTR analysis. Performing SSD on the PCs of the data is thus ideal within the industrial context.

The PCs were obtained as described in procedure 3.2.1. Based on Figure 47, it is clear that a scree test would not be effective. It is rather decided to retain 6 PCs, retaining approximately 90 % of the variance. The process seemed to experience inertia over long durations (auto correlated noise), thus SSD tuning parameters shown in Table 18 are implemented.

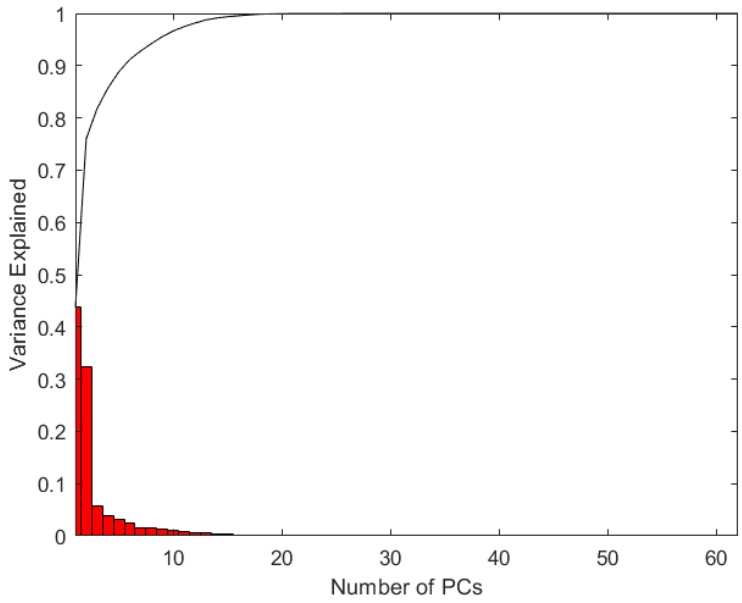


Figure 47: Pareto Chart of the milling circuit dataset displaying how the PCs explain the variance

The shifting window is implemented here due to its improved robustness. The visual results of the SSD analysis can be seen in Figure 48. Based on the SSD results the longest stationary period lasted 370 hours whereas the longest transient lasted 68 hours (excluding the initial transient resulting from the nature of the algorithm). Further, based on the analysis approximately 20 % of the dataset should be considered to be transient.

Table 18: Shifting window SSD parameters for mill circuit

Window Size (hours)	Variables	Threshold (θ_{ss})	Significance (α)	Delay
100	6 PCs	0.6	0,01%	0

It is clear that stationarity analysis can give some useful insight into the properties of a dataset, however the accuracy of the insight depends on the quality of the tuning parameters. Comparing Figure 48 and Figure 46 it seems that some of the same patterns in transitions vs. set point changes exist. However for reasons discussed in 4.1.2.6, it is decided not use SSD as a data filter prior to the state analysis procedure (3.2.3).

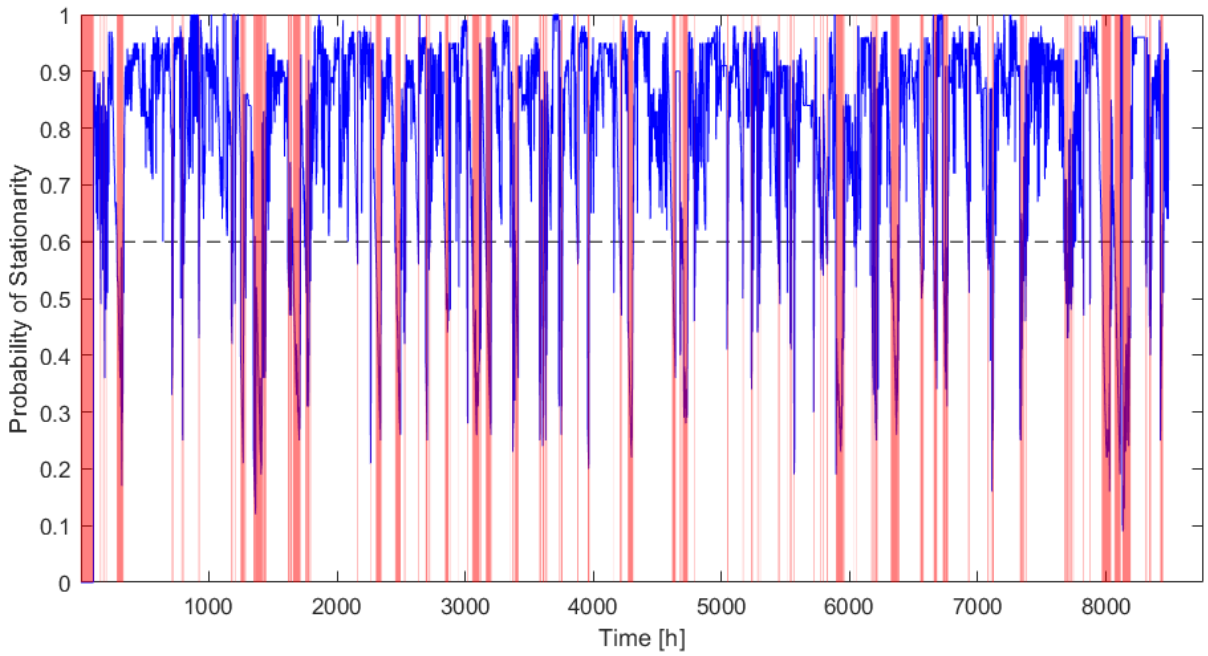


Figure 48: Sliding Window SSD results obtained from tuning in Table 18, where the red shaded regions denote detected transient periods. The blue data points describe the probability of a time window being stationary and the dotted line denotes Θ_{ss}

5.3 State Based Analysis on Industrial Data

Here the state based procedure 3.2.3 is performed on the 6 PCs obtained from the milling circuit data. It is extremely important to note that no expert knowledge on the dynamics of the process are known, this is a major inconvenience. As stated in previous sections, knowledge of the system is of crucial importance when providing number of modes, initialisation parameters and determining how to refine the GMM. The BIC results for the secondary mill data can be seen in Figure 49 a), where a GMM configuration containing 15 Gaussians is determined to be the best. Although it may be thought that selection a GMM configuration with for example 24 Gaussians may be better, a larger number of Gaussians also increase the complexity of subsequent refining procedures. It should therefore be clear that as the number of fit Gaussians increases, so does the network complexity and as a result decisions to merge modes or not becomes more difficult. The unrefined connectivity network can be seen in Appendix C Figure 63, which clearly indicates that the majority of the “heavy” Gaussians lie relatively close to each other. Figure 49 b) shows the Gaussian weights of the various Gaussians in descending order.

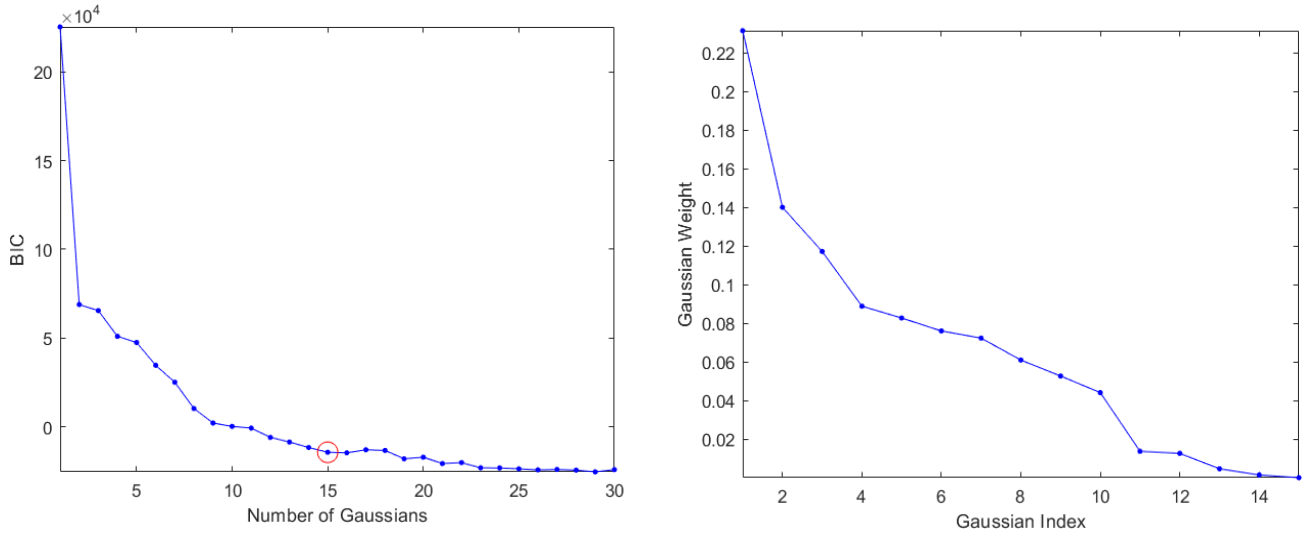


Figure 49: a) BIC for milling circuit data b) Gaussian weights of the GMM if 15 Gaussians are fit

Within this analysis it was decided not to perform Gaussian merging based on sequential analysis since the dynamics of the process are completely unknown. It is only recommended to perform Gaussian merging based on the time sequence if a clear idea of the mode switching conditions within the process are known. Based on Figure 49 b) it is assumed that Gaussians 11 until 15 contain transient data only, the importance of the clusters/modes is based purely on their Gaussian weight. Throughout the various investigations on simulated and industrial data, the need for unsupervised cluster quality metrics has become very apparent, less so when SSD can effectively be performed. Metrics such as multivariate skewness and kurtosis could be implemented to determine the normality of the data assigned to the various Gaussians. Thus, the decision to remove Gaussians will not be purely based on their weight.

The final obtained state time series is shown in Figure 51(from training data), here it should be noted that the state index is based on the descending order of the Gaussian weight. Although Figure 51 gives a useful indication of the periods during which a process was in which state, it is difficult seeing the switching conditions between the states. Here the connectivity diagram seen in Figure 50 (determined as in 3.2.3 step 8 ii) gives more useful insight.

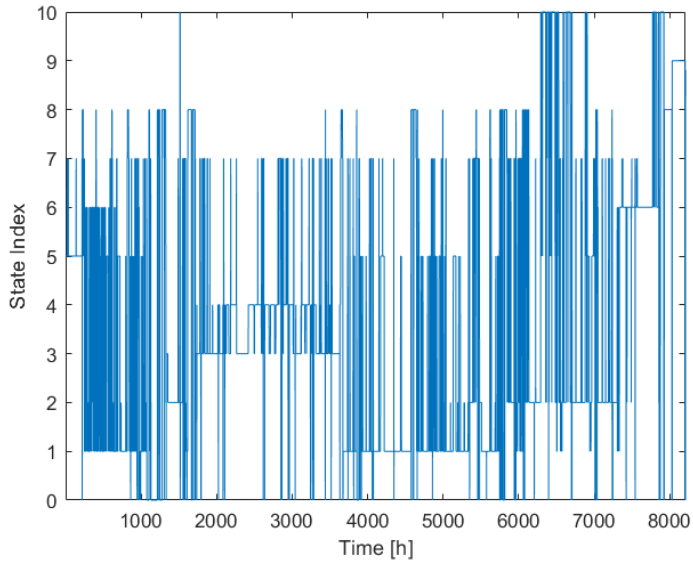


Figure 51: Secondary mill state with time, where state zero indicates transients

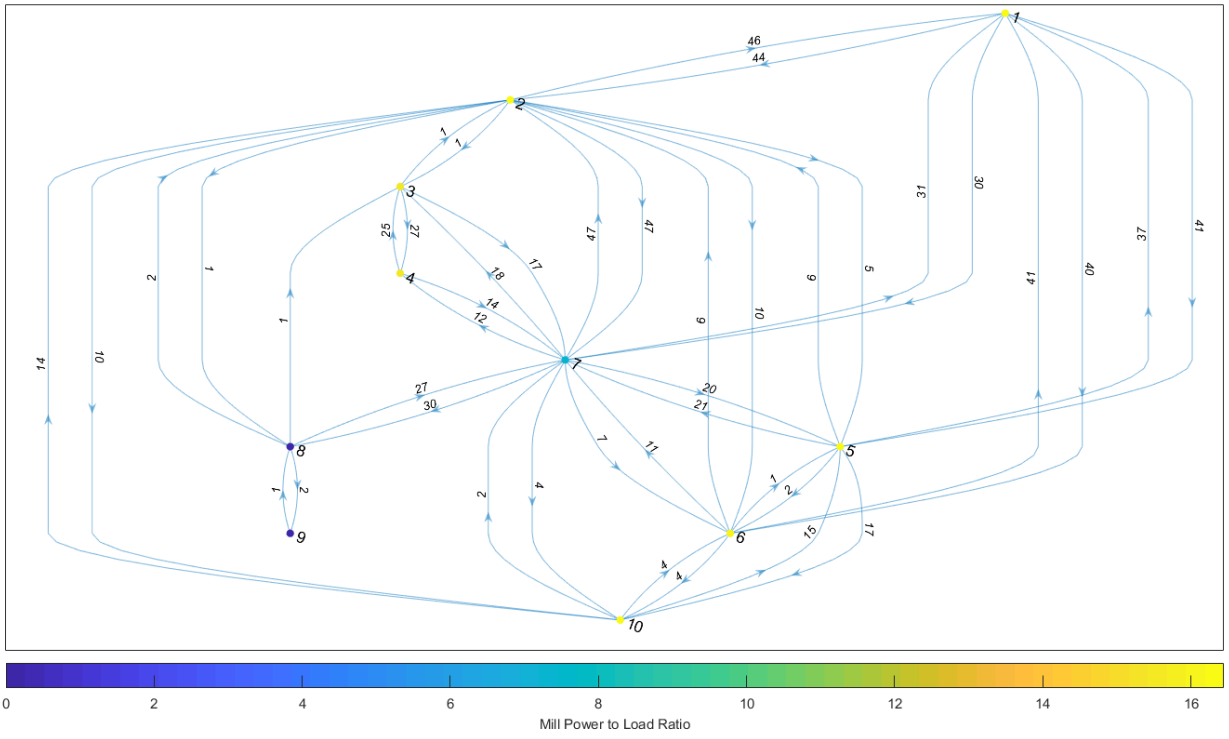


Figure 50: Connectivity diagram of the mode switches occurring within the milling circuit, where the node color indicates the mill power to load ratio

It should also be noticed that that the node colour in Figure 50 displays the mill power to load ratio, thus conveniently summarising key information that could assist with decision making. Ideally the particle size of the mill discharge would be assigned to each mode, this data was however not available. The states within the PC dimensions can be seen in Figure 62 in Appendix C, here it is clear that the modes are not effectively visualised (due to outliers/transient data). If however all assumed transient data is removed from the original high dimensional space and PCA is performed on that data, the previously detected modes are far more visible in Figure 52 a).

The state analysis seems to be quite effective at segmenting the modes. Based on the switching conditions seen in Figure 50 and the vicinity of the modes from each other in Appendix C Figure 63 (based on Euclidean distance), it is assumed that modes 1,2 and 6 are in fact the same mode. The merged results in the PC space can be seen in Appendix C Figure 61. Further, based on the switching conditions in Figure 50 it could also be assumed that modes 1 and 5 are the same, however in Appendix C Figure 63 it can be clearly seen that these modes are dissimilar (based on Euclidean distance) from one another.

Unlike the CSTR simulation dataset, it is clear that for the Industrial dataset considered here (milling circuit) the modes are far less distinct from one another. That is the variance resulting from a mode transition relative to the within mode variance is minor, making the analysis more difficult. Process drift further exacerbates this phenomena. For cases such as those discussed for modes 1,2 and 6, it is recommended that a further “in depth” analysis is performed. This “in depth” analysis entails performing PCA on purely those modes (high dimensional space) and following the same procedures of section 3.2.3. Thus an improved “resolution” of the modes could be achieved. Selection of adequate input data prior to PCA using expert knowledge of the process would also definitely improve all procedures, since unnecessary variables may mask data patterns.

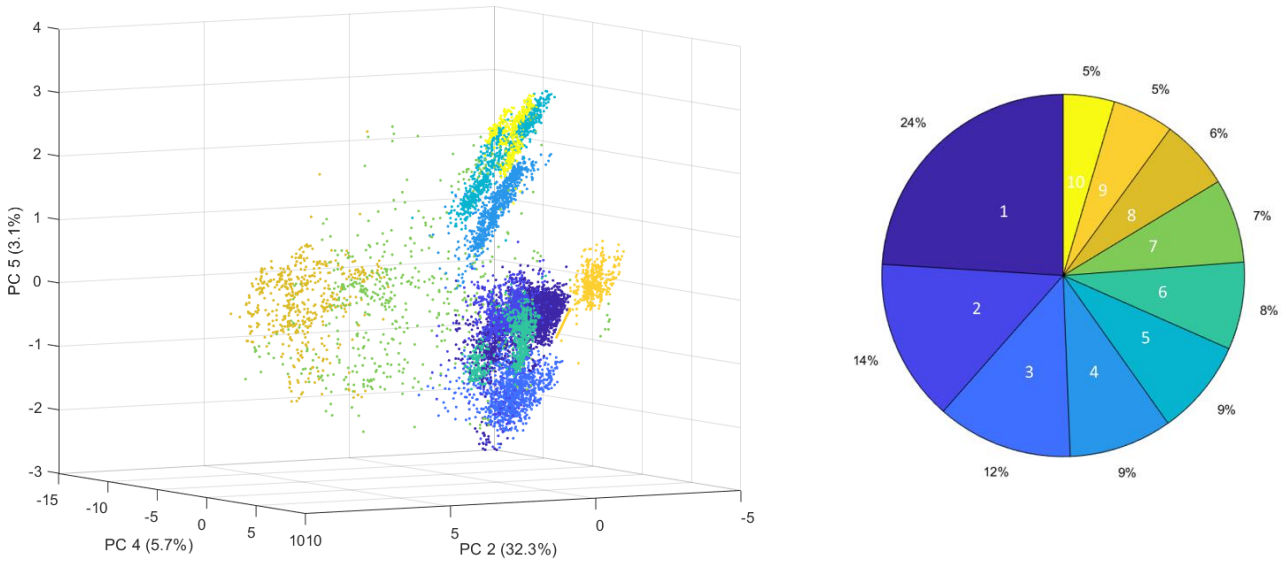


Figure 52: a) PC space that separates most detected modes b) Pie chart describing the percentage of the entire duration the circuit was in the corresponding mode

5.4 Causal Analysis for Mode Shifts

As discussed in section 5.1, procedure 3.2.4 is not applicable to the milling circuit as a result of an adaptive control implementation. Alternative methods of determining the "causes" of a process shifts therefore have to be determined. In the case of this investigation, methods of contribution plots are utilised. Here it is assumed that the normal mode of operation is the "heaviest" (most frequently occurring) mode within the data (ie. mode 1).

Using PCA to determine the subspace that best describes the normal mode of operation (procedure 3.2.1 only performed on mode 1), the contributions of each variable resulting in the mode shift from the "normal" can be determined. Figure 53 shows the total error contributions of the most important 20 variables (over the "abnormal" mode duration) that may have resulted in the mode transitions (from normal to the specific mode). The squared prediction error is determined by implementing Equations 2-19.

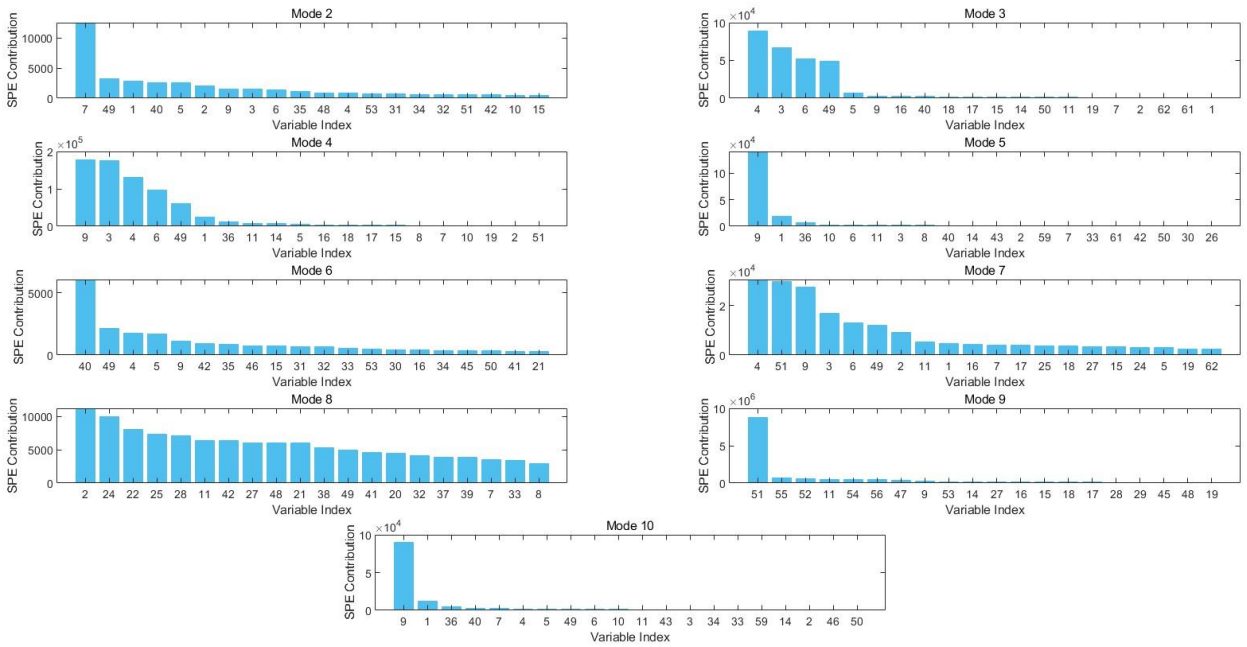


Figure 53: Variable contributions to the squared prediction error (SPE) of the various "abnormal" modes details of the variables indices can be found in Appendix C Table 27

For example, the 30 transitions from mode 1 to mode 7 may have resulted due to changes in the mill recycle stream from the sump, similarly mode 8 occurs due to changes in the mill power to load ratio. It is thus assumed that modes 7 and 8 are related to process shutdowns, mode 7 most likely being a transient state. Comparison of the contribution plots of mode 5 and mode 10 in Figure 53, it is deemed that these modes should most likely also be merged (Appendix C Figure 63 corresponds with this observation). These contribution plots can thus not only be used in GMM refining decisions, but can also be used to provide decision support during process operation. Information from Figure 50 could be combined with contribution plots, providing a useful tool for visualizing variables involved in specific transitions as well as mode shifting constraints. However, as discussed in 2.3.2.2, contribution plots don't distinguish transition causes from "symptoms" (Aldrich and Auret, 2013). Further, time lags between

process units were also not taken into account during this analysis. Data samples of the various process unit variables may thus not be synchronised, thus time delays between the causes and effects of transitions may exist (Aldrich and Auret, 2013). More sophisticated techniques and expert knowledge of the process topology is crucial for the determination of the root cause of a transition and the time delays within an actual process, only then can an effective process state map be determined.

Similarly, controller set point logs or datasets can be leveraged to determine the cause of a transition, as in this investigation. If however multiple process units are considered as within this industrial investigation, analysis of controller set point data becomes high dimensional in itself. If, however, a state-based analysis is performed on the controller data itself (as with mode discovery), “complex” controller states (rather multivariate distributions than specific set point values) could be assigned to specific process modes.

6 CONCLUSION

Within this investigation, a state-based decision support system for continuous multimodal processes is developed, which assists supervisors with process navigation, such that more optimal process conditions can be achieved. With the use of PCA, steady state detection, K-means clustering, and GMMs, modes are effectively discovered within historical simulated multimodal CSTR data. Key performance indicators are assigned to the discovered modes, based on the CSTR profitability. Set point datasets are leveraged to identify the causes of transients within the historical simulated data. Knowledge of the discovered modes, their KPIs, and switching conditions are then used to construct a process map. Further, Gaussian mixture modelling also serves as an ideal technique to model the linear relationship among variables as well as the operating points of the various discovered modes, such that modes can be identified within testing data. Thus, on simulated process data, the final state-based decision support system is effectively able to identify the current mode of operation, its optimality, the optimal process conditions (based on historical data), and the procedures required to transition to the optimal mode. Finally, the overall approach is evaluated on actual milling circuit process/set point data. Although the mode discovery procedure seems to work effectively without extensive process knowledge, it is determined that the "diagnosis" of switching conditions is difficult, thus the developed procedure may not be adequate. A more advanced approach taking into account the topology of a process may be required for the effective "diagnosis" of switching conditions within complex actual processes.

Based on literature it was hypothesized that steady state detection would be the ideal approach of dealing with transient states within data (Quiñones-Grueiro, Prieto-Moreno and Verde, 2019). In this investigation, it is determined that steady state detection is effective at removing transients from the multimodal CSTR data. However, determining effective hyper-parameter settings may be difficult to achieve and will require extensive process knowledge. A single SSD hyper-parameter setting may also not be effective for the varying dynamics that may occur within process data. Most importantly, extensive hierarchical process sectioning is required prior to steady state detection. This is due to the multivariate extension of the steady state detection technique, stating that an entire process is transient if a single variable is transient. Plant or even section wide stationarity is a demanding condition that is rarely met within the industrial context. The unique application of steady state detection on principal components within this investigation could, however, somewhat alleviate this implementation issue. Based on the steady state detection parameter settings applied to the milling circuit, it is determined that the mill was within a transient state 20 % of the time, information which could be useful for further analyses.

Within this investigation, it is determined that failing to remove transient data from the analysis, results in a model that less effectively identifies modes within testing data. A novel GMM refinement procedure is developed, which can effectively simplify a process map as well as remove transient data from the analysis. By leveraging the switching frequency between modes, the Euclidean distance between mode means, and their Gaussian weight, Gaussians can be merged or removed from the model. This refinement procedure can be used in conjunction with steady state detection or as an alternative, such that transient data can be effectively excluded from the state analysis. However, within the context of complex industrial processes, the refinement procedure is more reliable, if it is performed with the guidance of

expert knowledge. Further, a novel state visualization approach is developed, projecting high dimensional multimodal data into a two-dimensional space. Human-machine interactions are therefore maximised, allowing supervisors to effectively navigate processes. This same connectivity visualisation approach is especially useful when refining GMMs. Thus, this work provides useful techniques for the discovery of modes within historical process data (which is often not thoroughly addressed in literature) and the identification of modes within real-time data. Future research should leverage these techniques to more effectively "diagnose" causes of transitions between modes.

Ultimately, the described state-based decision support system assimilates real-time measurements, projects these measurements to a specific state, indicates the path of the operating point in the state space in real-time, and identifies the most important variables which are responsible for the operational state change (Wang, 1999). This creates a virtual environment that allows for comprehensible assessment of process performance and the factors which determine it, which will provide guidance to supervisors by creating useful information from data that will ultimately help in decision making (Wang, 1999). Faster and more effective decision making concerning operating mode shifts will form a key strategic tool for enhancing business competition (Wang, 1999).

7 RECOMMENDATIONS

The described state-based system does not take into account process deviation or drift. For certain processes, this may be a serious issue. Implementations of adaptive techniques within the suggested approach could be useful (Addo, 2019). The addition of new modes to the model was also not considered, in a sense this also corresponds to the adaptive extension of the model. If stationarity analysis can effectively be applied to a dataset, it will be especially useful here (ie. a set of samples are not identified to be any of the known modes but are not transient, then a new mode should most likely be added to the model).

Many more advanced clustering algorithms have been developed that could be used to provide initial parameter estimates to the EM of the GMM (Thomas, Zhu and Romagnoli, 2018). Locality preserving projections or robust extensions of PCA may in certain cases be more effective at reducing dimensionality than "vanilla" PCA (Yu, 2016). Sometimes it may be useful to perform a more in-depth analysis on specific modes. That is performing the state-based analysis approach on specific modes, improved detection resolutions could be achieved in this manner. Additional cluster evaluation metrics can be utilised during GMM refining, the multivariate skewness and kurtosis of data assigned to a cluster may be useful. The probability density (for example the maximum $NLLP$ of a Gaussian) of a Gaussian may also be a useful metric indicating the extent of variance it models. Gaussians which model large variance most likely fit transient data.

Further, incorporating time lags and a processes' topology will definitely improve the decision support system (Aldrich and Auret, 2013). Linking controller states (for example performing the state analysis on set point datasets) to process states may give a better indication of the switching conditions required to shift a mode.

8 REFERENCES

Addo, P. (2019) *Adaptive Process Monitoring using Principal Component Analysis and Gaussian Mixture Models by*. Stellenbosch University.

Afzal, M. S., Tan, W. and Chen, T. (2017) 'Process monitoring for multimodal processes with mode-reachability constraints', *IEEE Transactions on Industrial Electronics*. IEEE, 64(5), pp. 4325–4335. doi: 10.1109/TIE.2017.2677351.

Aldrich, C. *et al.* (2014) 'Visualization of the controller states of an autogenous mill from time series data', *Minerals Engineering*. Elsevier Ltd, 56, pp. 1–9. doi: 10.1016/j.mineng.2013.10.018.

Aldrich, C. and Auret, L. (2013) *Unsupervised Process Monitoring and Fault Diagnosis with Machine Learning Methods*, *Advances in Computer Vision and Pattern Recognition*. Springer. doi: 10.1007/978-1-4471-5185-2.

AlGhazzawi, A. and Lennox, B. (2008) 'Monitoring a complex refining process using multivariate statistics', *Control Engineering Practice*, 16(3), pp. 294–307. doi: 10.1016/j.conengprac.2007.04.014.

Arthur, D. (2007) 'k-means ++ : The Advantages of Careful Seeding', in *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*.

Ashleigh, M. J. and Stanton, N. A. (2001) 'Trust: Key Elements in Human Supervisory Control Domains', *Cognition, Technology & Work*, 3(2), pp. 92–100. doi: 10.1007/pl00011527.

Brown, P. R. and Rhinehart, R. R. (2000) 'Automated steady-state identification in multivariable systems', *Hydrocarbon Processing*, 79(9), pp. 79–83.

Chen, J. and Howell, J. (2001) 'A self-validating control system based approach to plant fault detection and diagnosis', 25, pp. 337–358.

Chiang, L. and Russell, E. L. (2001) *Fault Detection and Diagnosis in Industrial Systems*. Springer-Verlag London.

Choi, S. W., Park, J. H. and Lee, I. B. (2004) 'Process monitoring using a Gaussian mixture model via principal component analysis and discriminant analysis', *Computers and Chemical Engineering*, 28(8), pp. 1377–1387. doi: 10.1016/j.compchemeng.2003.09.031.

Cummings, M. L., Bruni, S. and Mitchell, P. J. (2010) 'Human Supervisory Control Challenges in Network-Centric Operations', *Reviews of Human Factors and Ergonomics*, 6(1), pp. 34–78. doi: 10.1518/155723410x12849346788660.

Ding, C. (2006) 'R 1 -PCA : Rotational Invariant L 1 -norm Principal Component Analysis for Robust Subspace Factorization', in *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pp. 281–288. doi: https://dl.acm.org/doi/10.1145/1143844.1143880.

Figueiredo, M. A. T., Member, S. and Jain, A. K. (2002) 'Unsupervised Learning of Finite Mixture Models', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3), pp. 381–396.

Fogler, H. S. (2018) *Essentials of Chemical Reaction Engineering*. 2nd edn. Prentice Hall.

Ha, D. *et al.* (2017) 'Multi-mode operation of principal component analysis with k-nearest neighbor algorithm to monitor compressors for liquefied natural gas mixed refrigerant processes', *Computers and Chemical Engineering*. Elsevier Ltd, 106, pp. 96–105. doi: 10.1016/j.compchemeng.2017.05.029.

Halligan, S., Altman, D. G. and Mallett, S. (2015) 'Disadvantages of using the area under the receiver operating characteristic curve to assess imaging tests: A discussion and proposal for an alternative approach', *European Radiology*, 25(4), pp. 932–939. doi: 10.1007/s00330-014-3487-0.

James, G. *et al.* (2013) *An Introduction to Statistical Learning*. New York, NY: Springer New York (Springer Texts in Statistics). doi: 10.1007/978-1-4614-7138-7.

Jiang, Y., Li, K. and Yin, S. (2018) 'Cyber-physical system based factory monitoring and fault diagnosis framework with plant-wide performance optimization', *Proceedings - 2018 IEEE Industrial Cyber-Physical Systems, ICPS 2018*, pp. 240–245. doi: 10.1109/ICPHYS.2018.8387666.

Jolliffe, I. T. (2002) 'Principal Component Analysis, Second Edition', *Encyclopedia of Statistics in Behavioral Science*, 30(3), p. 487. doi: 10.2307/1270093.

Kelly, J. D. and Hedengren, J. D. (2013) 'A steady-state detection (SSD) algorithm to detect non-stationary drifts in processes', *Journal of Process Control*, 23(3), pp. 326–331. doi: 10.1016/j.jprocont.2012.12.001.

Kruger, U. and Xie, L. (2012) *Statistical Monitoring of Complex Multivariate Processes*, *Statistics in Practice*. Wiley. doi: 10.1002/9780470517253.

Kuespert, D. R. and McAvoy, T. J. (1994) 'Knowledge extraction in chemical process control', *Chemical Engineering Communications*, 130(1), pp. 251–264. doi: 10.1080/00986449408936279.

Liu, J. and Chen, D. S. (2010) 'Nonstationary fault detection and diagnosis for multimode processes', *AIChE Journal*, 56(1), pp. 207–219. doi: 10.1002/aic.11999.

Liu, Y., Wang, F. and Chang, Y. (2015) 'Operating optimality assessment and nonoptimal cause identification for non-Gaussian multimode processes with transitions', *Chemical Engineering Science*. Elsevier, 137, pp. 106–118. doi: 10.1016/j.ces.2015.06.016.

Liu, Y., Wang, F. and Chang, Y. (2016) 'Operating optimality assessment based on optimality related variations and nonoptimal cause identification for industrial processes', *Journal of Process Control*. Elsevier Ltd, 39, pp. 11–20. doi: 10.1016/j.jprocont.2015.12.008.

Liu, Y., Wang, F. and Chang, Y. (2018) 'Operating performance assessment based on GMM-GPR for gold hydrometallurgy processes', *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*. IEEE, pp. 3222–3226. doi: 10.1109/CCDC.2018.8407679.

Mansour, M. and Ellis, J. E. (2008) 'Methodology of on-line optimisation applied to a chemical reactor', *Applied Mathematical Modelling*, 32, pp. 170–184. doi: 10.1016/j.apm.2006.11.014.

Mardia, K. V. (1970) 'Measures of Multivariate Skewness and Kurtosis with Applications', *Biometrika*, 57(3), p. 519. doi: 10.2307/2334770.

Marlin, T. (1995) *Process Control: Designing Processes and Control Systems for Dynamic Performance*, *Technology & Engineering*. McGraw-Hill Education. doi: https://www.amazon.com/Process-Control-Designing-Processes-Performance/dp/0070393621.

McLeod, R. W. (2015) *Designing for Human Reliability*. 1st edn, *Human Factors Engineering in the Oil, Gas, and Process Industries*. 1st edn. Elsevier Ltd. doi: 10.1016/c2014-0-02149-1.

Miskin, J. (2016) *Control performance assessment for a high pressure leaching process by means of fault database creation and simulation*. Stellenbosch University.

Morari, M. and Stephanopoulos, G. (1980) 'Studies in the Synthesis of Control Structures for Chemical Processes: Part IV. Design of Steady-State Optimizing Control Structures for Chemical Process Units', *AIChE Journal*, 26(2), pp. 232–246. doi: 10.1002/aic.690260206.

Ng, Y. S. and Srinivasan, R. (2008) 'Multivariate temporal data analysis using self-organizing Maps. 2. Monitoring and diagnosis of multistate operations', *Industrial and Engineering Chemistry Research*, 47(20), pp. 7758–7771. doi: 10.1021/ie071022y.

Nimmo, I. (1993) 'Start up plants safely', *Chemical Engineering Progress*, 89(12), pp. 66–69.

Nimmo, I. (1995) 'Adequately Address Abnormal Operations', *Chemical engineering progress*, 91(9), p. 36.

Quiñones-Grueiro, M., Prieto-Moreno, A. and Verde, C. (2019) 'Data-driven monitoring of multimode

continuous processes: A review', *Chemometrics and Intelligent Laboratory Systems*, 189(December 2018), pp. 56–71. doi: 10.1016/j.chemolab.2019.03.012.

Rhinehart, R. R. (2013) 'Automated steady and transient state identification in noisy processes', *Proceedings of the American Control Conference*, pp. 4477–4493. doi: 10.1109/acc.2013.6580530.

Salvendy, G. (2012) *Handbook of Human Factors and Ergonomics: Fourth Edition*, *Handbook of Human Factors and Ergonomics: Fourth Edition*. doi: 10.1002/9781118131350.

Sasaki, Y. (2007) *The truth of the F-measure*, *Toyota Technological Institute*.

Schwarz, G. (1978) '"Estimating the Dimension of a Model."', *The Annals of Statistics*, 6(2), pp. 461–464. doi: 10.2307/2958889.

Seborg, D. E., Edgar, T. and Mellichamp, D. (2004) *Process Dynamics and Control*. 2nd edn. Edited by B. Zobrist. Wiley.

Sebzalli, Y. M., Li, R. F. and Chen, F. Z. (2000) 'Knowledge discovery from process operational data for assessment and monitoring of operator ' s performance', *Computers & Chemical Engineering*, 24, pp. 409–414.

Sebzalli, Y. M. and Wang, X. Z. (2001) 'Knowledge discovery from process operational data using PCA and fuzzy clustering', *Engineering Applications of Artificial Intelligence*, 14(5), pp. 607–616. doi: 10.1016/S0952-1976(01)00032-X.

Simon, D. L. and Litt, J. S. (2011) 'A Data Filter for Identifying Steady-State Operating Points in Engine Flight Data for Condition', *Journal of Engineering for Gas Turbines and Power*, 133(7), pp. 1–8. doi: 10.1115/1.4002318.

Slišković, D., Grbić, R. and Hocenski, Ž. (2012) 'Multivariate statistical process monitoring', *Tehnicki Vjesnik*, 19(1), pp. 33–41.

Smarra, F., Jain, A. and de Rubeis, T. (2018) 'Data-driven model predictive control using random forests for building energy optimization and climate control', *Applied Energy*. Elsevier, 226(September 2017), pp. 1252–1272. doi: 10.1016/j.apenergy.2018.02.126.

Song, B., Tan, S. and Shi, H. (2016) 'Key principal components with recursive local outlier factor for multimode chemical process monitoring', *Journal of Process Control*. Elsevier Ltd, 47, pp. 136–149. doi: 10.1016/j.jprocont.2016.09.006.

Srinivasan, R., Viswanathan, P. and Vedam, H. (2005) 'A framework for managing transitions in chemical plants', *Computers and Chemical Engineering*, 29(2), pp. 305–322. doi: 10.1016/j.compchemeng.2004.09.024.

Srinivasan, R., Wang, C. and Ho, W. K. (2004) 'Dynamic Principal Component Analysis Based Methodology for Clustering Process States in Agile Chemical Plants', *Industrial & Engineering Chemistry Research*, 43(9), pp. 2123–2139. doi: 10.1021/ie034051r.

Thissen, U., Swierenga, H. and De Weijer, A. P. (2005) 'Multivariate statistical process control using mixture modelling', *Journal of Chemometrics*, 19(1), pp. 23–31. doi: 10.1002/cem.903.

Thomas, M. C., Zhu, W. and Romagnoli, J. A. (2018) 'Data mining and clustering in chemical process databases for monitoring and knowledge discovery', *Journal of Process Control*. Elsevier Ltd, 67, pp. 160–175. doi: 10.1016/j.jprocont.2017.02.006.

Wang, X. Z. (1999) *Data Mining and Knowledge Discovery for Process Monitoring and Control*. London: Springer London (Advances in Industrial Control). doi: 10.1007/978-1-4471-0421-6.

Wang, X. Z., Chen, B. H. and Yang, S. H. (1997) 'Fuzzy rule generation from data for process operational decision support', *Computers and Chemical Engineering*, 21(SUPPL.1). doi: 10.1016/s0098-1354(97)87578-3.

Wang, Xiaoyang, Wang, Xin and Wang, Z. (2013) 'A novel method for detecting processes with multi-state modes', *Control Engineering Practice*. Elsevier, 21(12), pp. 1788–1794. doi: 10.1016/j.conengprac.2013.08.016.

Winston, W. (2003) *Operations Research*, *Data Handling in Science and Technology*. doi: 10.1016/S0922-3487(08)70238-7.

Wolpert, D. and Macready, W. (1995) 'No Free Lunch Theorems for Optimization', *IEEE Transactions on Evolutionary Computation*, 1(67).

Xie, X. and Shi, H. (2012) 'Dynamic multimode process modeling and monitoring using adaptive gaussian mixture models', *Industrial and Engineering Chemistry Research*, 51(15), pp. 5497–5505. doi: 10.1021/ie202720y.

Xiong, H., Wu, J. and Chen, J. (2009) 'K-means clustering versus validation measures: A data-distribution perspective', *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(2), pp. 318–331. doi: 10.1109/TSMCB.2008.2004559.

Xu, H., Wu, J. and Tseng, T. L. B. (2018) 'An efficient method for online identification of steady state for multivariate systems', *ASME 2018 13th International Manufacturing Science and Engineering Conference, MSEC 2018*, 4(2), pp. 1–9. doi: 10.1115/MSEC2018-6565.

Ye, L., Liu, Y. and Fei, Z. (2009) 'Online probabilistic assessment of operating performance based on safety and optimality indices for multimode industrial processes', *Industrial and Engineering Chemistry Research*, 48(24), pp. 10912–10923. doi: 10.1021/ie801870g.

Ying, Y., Li, Z. and Yang, M. (2020) 'Multimode operating performance visualization and nonoptimal cause identification', *Mdpi Processes*, 8(1). doi: 10.3390/pr8010123.

Yoon, S. and Macgregor, J. F. (2001) 'Fault diagnosis with multivariate statistical models part I: using steady state fault signatures', *Journal of Process Control*, 11(4), pp. 387–400. doi: 10.1016/S0959-1524(00)00008-1.

Yu, J. (2011) 'Fault detection using principal components-based gaussian mixture model for semiconductor manufacturing processes', *IEEE Transactions on Semiconductor Manufacturing*. IEEE, 24(3), pp. 432–444. doi: 10.1109/TSM.2011.2154850.

Yu, J. (2016) 'Process monitoring through manifold regularization-based GMM with global/local information', *Journal of Process Control*. Elsevier Ltd, 45, pp. 84–99. doi: 10.1016/j.jprocont.2016.07.006.

Yu, J. and Qin, S. J. (2008) 'Multimode process monitoring with bayesian inference-based finite Gaussian mixture models', *AIChE Journal*, 54(7), pp. 1811–1829. doi: 10.1002/aic.11515.

Zhang, P. (2010) *Advanced Industrial Control Technology*. 1st edn. Elsevier Ltd. doi: 10.1016/C2009-0-20337-0.

Zhang, S. *et al.* (2015) 'Novel Monitoring Strategy Combining the Advantages of the Multiple Modeling Strategy and Gaussian Mixture Model for Multimode Processes', *Industrial & Engineering Chemistry Research*, 54(47). doi: 10.1021/acs.iecr.5b00373.

Zhang, Y., Li, S. and Teng, Y. (2012) 'Dynamic processes monitoring using recursive kernel principal component analysis', *Chemical Engineering Science*. Elsevier, 72, pp. 78–86. doi: 10.1016/j.ces.2011.12.026.

Zhao, S. J., Zhang, J. and Xu, Y. M. (2004) 'Multiple Principle Component Analysis Models', *Ind. Eng. Chem. Res.*, 43, pp. 7025–7035.

APPENDIX A: SIMPLE CSTR DATA DESCRIPTION AND APPROACH PERFORMANCE

Table 19: Noiseless Input Variable Pairings for the Dataset used in stationarity analysis and training data (seed = 60, Duration = 8000 hours)

Time [h]	Level SP [m]	CSTR Temperature SP [K]	Inlet Temperature [K]	Cooling Water Temperature [K]	Methanol Flowrate [kmol/h]	Reactant Water Flowrate [kmol/h]
0	0,98	305	279	271	45,36	453,59
571,6	0,78	305	279	271	45,36	453,59
882,4	0,98	305	279	271	45,36	453,59
1158,3	0,98	305	279	271	36,29	453,59
1402	0,98	305	279	271	36,29	453,59
1632,8	0,98	310	279	271	36,29	453,59
1942	0,98	310	279	271	36,29	453,59
2193,9	0,98	310	279	271	36,29	453,59
2453,1	0,98	310	279	271	36,29	453,59
2838,6	0,98	310	279	271	36,29	453,59
3075,7	0,98	310	279	271	36,29	453,59
3297,1	0,98	310	279	271	36,29	453,59
3625,6	0,98	321	279	271	36,29	453,59
3866,1	0,98	321	279	271	36,29	453,59
4179,9	0,98	321	279	271	36,29	453,59
4442	0,98	321	279	271	36,29	453,59
4825,8	0,98	321	279	271	36,29	453,59
5035,5	0,78	321	279	271	36,29	453,59
5335,6	0,98	321	279	271	36,29	453,59
5640	0,98	321	279	271	36,29	453,59
5911,1	0,98	321	279	271	36,29	453,59
6166,1	0,98	321	279	271	36,29	453,59
6417,5	0,98	321	279	271	36,29	453,59
6626,7	0,78	321	279	271	36,29	453,59
6950,8	0,78	321	279	271	36,29	453,59
7191,3	0,78	321	279	271	36,29	453,59
7522,6	0,78	321	279	271	36,29	453,59
7852,1	0,78	321	279	271	36,29	453,59

Table 20: Noiseless Input Variable Pairings for the Dataset used for Testing data (seed = 60, Duration = 8000 hours)

Time [h]	Level [m]	CSTR Temperature [K]	Inlet Temperature [K]	Cooling Water Temperature [K]	Methanol Flowrate [kmol/h]	Reactant Water Flowrate [kmol/h]
230,9	0,98	323	297	289	45,36	453,59
540	0,78	323	297	289	45,36	453,59
791,9	0,98	323	297	289	45,36	453,59
1051,2	0,98	323	297	289	36,29	453,59
1436,6	0,98	323	297	289	36,29	453,59
1673,8	0,98	328	297	289	36,29	453,59
1895,1	0,98	328	297	289	36,29	453,59
2223,6	0,98	328	297	289	36,29	453,59
2464,1	0,98	328	297	289	36,29	453,59
2777,9	0,98	328	297	289	36,29	453,59
3040	0,98	328	297	289	36,29	453,59
3423,9	0,98	328	297	289	36,29	453,59
3633,5	0,98	339	297	289	36,29	453,59
3933,6	0,98	339	297	289	36,29	453,59
4238	0,98	339	297	289	36,29	453,59
4509,1	0,98	339	297	289	36,29	453,59
4764,1	0,98	339	297	289	36,29	453,59
5015,6	0,78	339	297	289	36,29	453,59
5224,8	0,98	339	297	289	36,29	453,59
5548,8	0,98	339	297	289	36,29	453,59
5789,4	0,98	339	297	289	36,29	453,59
6120,6	0,98	339	297	289	36,29	453,59
6450,1	0,98	339	297	289	36,29	453,59
6750,1	0,78	339	297	289	36,29	453,59
6980,7	0,78	339	297	289	36,29	453,59
7220,9	0,78	339	297	289	36,29	453,59
7598	0,78	339	297	289	36,29	453,59
7919,3	0,78	339	297	289	36,29	453,59

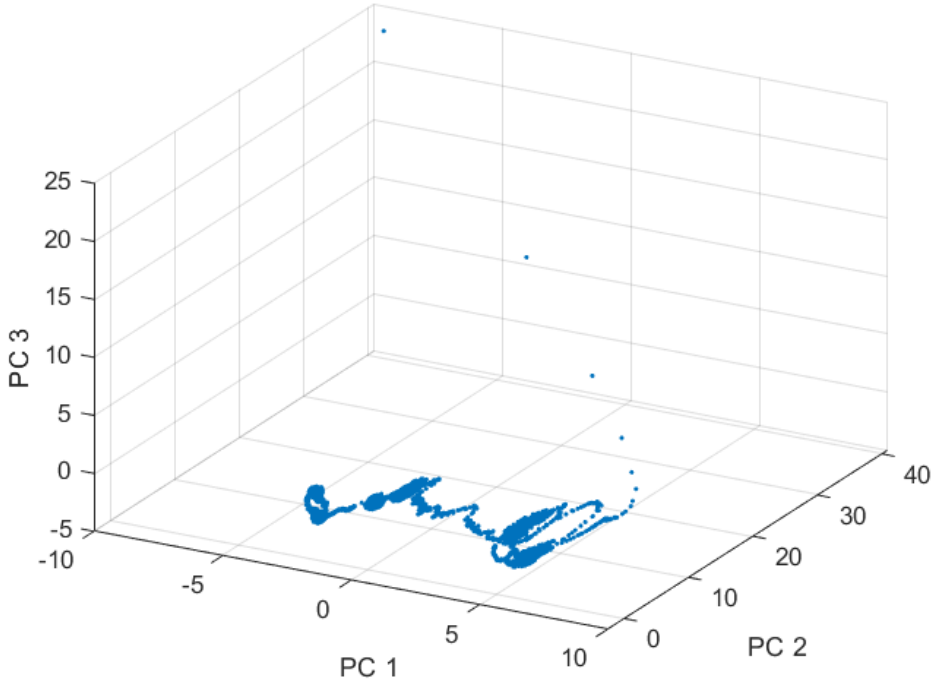


Figure 54: CSTR dataset including start-up in the PC space

Table 21: Performance metrics of the various approaches on the simple CSTR simulation

a)	F1	Precision	Recall	**b)**	F1	Precision	Recall	**c)**	F1	Precision	Recall
1	0.998	0.999	0.998	1	0.998	0.998	0.998	1	0.998	0.999	0.997
2	0.996	0.992	0.999	2	0.996	0.993	0.999	2	0.996	0.993	0.998
3	0.998	0.999	0.997	3	0.996	0.998	0.994	3	0.994	0.998	0.990
4	0.964	0.980	0.949	4	0.988	0.979	0.996	4	0.988	0.981	0.995
5	0.951	0.984	0.919	5	0.982	0.983	0.980	5	0.984	0.982	0.986
6	0.971	0.948	0.996	6	0.972	0.946	0.999	6	0.970	0.946	0.994
0	0.589	0.528	0.667	0	0.707	0.783	0.644	0	0.702	0.746	0.662
d)	F1	Precision	Recall	**e)**	F1	Precision	Recall	**f)**	F1	Precision	Recall
1	0.998	0.998	0.998	1	0.951	0.998	0.908	1	0.995	0.995	0.996
2	0.996	0.993	0.999	2	0.996	0.993	0.998	2	0.992	0.992	0.991
3	0.996	0.998	0.994	3	0.994	0.998	0.990	3	0.993	0.993	0.994
4	0.846	0.734	1.000	4	0.848	0.737	0.999	4	0.984	0.970	0.997
5	0.970	0.943	0.998	5	0.984	0.982	0.986	5	0.937	0.888	0.993
6	NaN	0.000	0.000	6	NaN	0.000	0.000	6	0.942	0.890	1.000
0	0.156	0.513	0.092	0	0.634	0.720	0.567	0	0.162	0.382	0.103
g)	F1	Precision	Recall								
1	0.995	0.995	0.995								
2	0.994	0.991	0.996								
3	0.991	0.994	0.988								
4	0.985	0.976	0.995								
5	0.979	0.978	0.980								
6	0.965	0.937	0.994								
0	0.558	0.620	0.508								

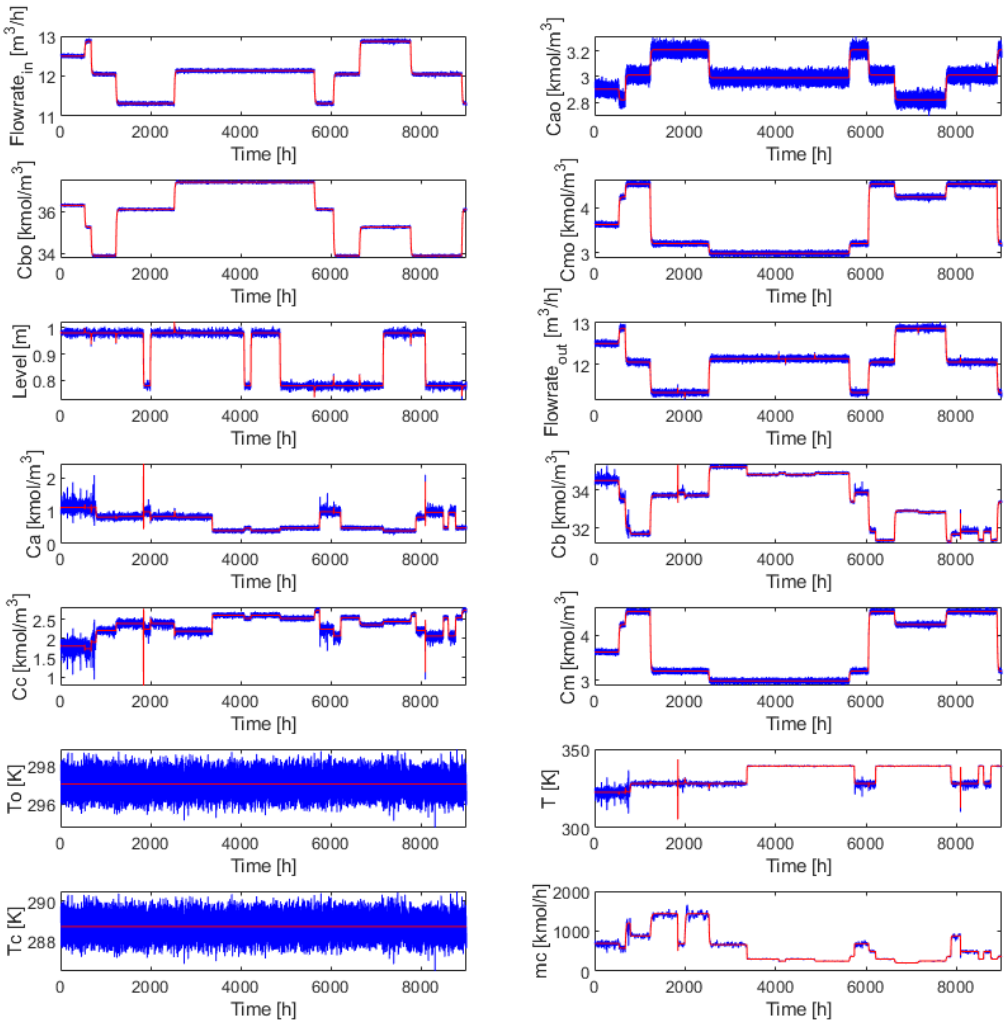
APPENDIX B: SIMULATED CSTR DATA AND APPROACH PERFORMANCE

Table 22: Noiseless Input Variable Pairings for the Dataset used as training data (seed = 50, Duration = 9000 hours, min steady state time = 100 hours)

Time [h]	Level [m]	CSTR Temperature [K]	Inlet Temperature [K]	Cooling Water Temperature [K]	Methanol Flowrate [kmol/h]	Reactant Water Flowrate [kmol/h]
0	0,98	323	297	289	45,36	453,59
349,9	0,98	323	297	289	45,36	453,59
534,6	0,98	323	297	289	54,43	453,59
676,6	0,98	323	297	289	54,43	408,23
785,9	0,98	328	297	289	54,43	408,23
984,3	0,98	328	297	289	54,43	408,23
1093	0,98	328	297	289	54,43	408,23
1229,5	0,98	328	297	289	36,29	408,23
1421,9	0,98	328	297	289	36,29	408,23
1552	0,98	328	297	289	36,29	408,23
1734,5	0,98	328	297	289	36,29	408,23
1837,9	0,78	328	297	289	36,29	408,23
1993,3	0,98	328	297	289	36,29	408,23
2131,2	0,98	328	297	289	36,29	408,23
2253	0,98	328	297	289	36,29	408,23
2368,5	0,98	328	297	289	36,29	408,23
2523	0,98	328	297	289	36,29	453,59
2649	0,98	328	297	289	36,29	453,59
2778,6	0,98	328	297	289	36,29	453,59
2971,3	0,98	328	297	289	36,29	453,59
3089,9	0,98	328	297	289	36,29	453,59
3200,6	0,98	328	297	289	36,29	453,59
3364,9	0,98	339	297	289	36,29	453,59
3485,1	0,98	339	297	289	36,29	453,59
3642	0,98	339	297	289	36,29	453,59
3773	0,98	339	297	289	36,29	453,59
3965	0,98	339	297	289	36,29	453,59
4069,8	0,78	339	297	289	36,29	453,59
4219,8	0,98	339	297	289	36,29	453,59
4372	0,98	339	297	289	36,29	453,59
4507,6	0,98	339	297	289	36,29	453,59
4635,1	0,98	339	297	289	36,29	453,59
4760,8	0,98	339	297	289	36,29	453,59
4865,4	0,78	339	297	289	36,29	453,59
5027,4	0,78	339	297	289	36,29	453,59
5147,7	0,78	339	297	289	36,29	453,59
5313,3	0,78	339	297	289	36,29	453,59
5478,1	0,78	339	297	289	36,29	453,59
5628,1	0,78	339	297	289	36,29	408,23
5743,4	0,78	328	297	289	36,29	408,23
5863,5	0,78	328	297	289	36,29	408,23
6052,1	0,78	328	297	289	54,43	408,23
6212,7	0,78	339	297	289	54,43	408,23
6384,6	0,78	339	297	289	54,43	408,23
6507,5	0,78	339	297	289	54,43	408,23
6629	0,78	339	297	289	54,43	453,59
6812,7	0,78	339	297	289	54,43	453,59
6985,9	0,78	339	297	289	54,43	453,59
7153,1	0,98	339	297	289	54,43	453,59
7307,5	0,98	339	297	289	54,43	453,59
7459,1	0,98	339	297	289	54,43	453,59
7618,9	0,98	339	297	289	54,43	453,59
7762,1	0,98	339	297	289	54,43	408,23
7876,9	0,98	328	297	289	54,43	408,23
7985,2	0,98	328	297	289	54,43	408,23
8086,1	0,78	328	297	289	54,43	408,23
8202,4	0,78	328	297	289	54,43	408,23
8325,7	0,78	328	297	289	54,43	408,23
8486,9	0,78	339	297	289	54,43	408,23
8598,5	0,78	328	297	289	54,43	408,23
8759,7	0,78	339	297	289	54,43	408,23
8897,2	0,78	339	297	289	36,29	408,23

Table 23: Noiseless Input Variable Pairings for the Dataset used as testing data (seed = 50, Duration = 9000 hours, min steady state time = 100 hours)

Time [h]	Level [m]	CSTR Temperature [K]	Inlet Temperature [K]	Cooling Water Temperature [K]	Methanol Flowrate [kmol/h]	Reactant Water Flowrate [kmol/h]
198,4	0,98	323	297	289	45,36	453,59
307,2	0,98	323	297	289	45,36	453,59
443,7	0,98	323	297	289	54,43	453,59
636,1	0,98	323	297	289	54,43	408,23
766,2	0,98	328	297	289	54,43	408,23
948,6	0,98	328	297	289	54,43	408,23
1052	0,98	328	297	289	54,43	408,23
1207,4	0,98	328	297	289	36,29	408,23
1345,3	0,98	328	297	289	36,29	408,23
1467,1	0,98	328	297	289	36,29	408,23
1582,6	0,98	328	297	289	36,29	408,23
1737,2	0,78	328	297	289	36,29	408,23
1863,1	0,98	328	297	289	36,29	408,23
1992,7	0,98	328	297	289	36,29	408,23
2185,5	0,98	328	297	289	36,29	408,23
2304	0,98	328	297	289	36,29	408,23
2414,7	0,98	328	297	289	36,29	453,59
2579	0,98	328	297	289	36,29	453,59
2699,2	0,98	328	297	289	36,29	453,59
2856,1	0,98	328	297	289	36,29	453,59
2987,2	0,98	328	297	289	36,29	453,59
3179,1	0,98	328	297	289	36,29	453,59
3283,9	0,98	339	297	289	36,29	453,59
3434	0,98	339	297	289	36,29	453,59
3586,2	0,98	339	297	289	36,29	453,59
3721,7	0,98	339	297	289	36,29	453,59
3849,2	0,98	339	297	289	36,29	453,59
3974,9	0,78	339	297	289	36,29	453,59
4079,5	0,98	339	297	289	36,29	453,59
4241,6	0,98	339	297	289	36,29	453,59
4361,8	0,98	339	297	289	36,29	453,59
4527,5	0,98	339	297	289	36,29	453,59
4692,2	0,98	339	297	289	36,29	453,59
4842,2	0,78	339	297	289	36,29	453,59
4957,5	0,78	339	297	289	36,29	453,59
5077,6	0,78	339	297	289	36,29	453,59
5266,2	0,78	339	297	289	36,29	453,59
5426,8	0,78	339	297	289	36,29	453,59
5598,7	0,78	339	297	289	36,29	408,23
5721,6	0,78	328	297	289	36,29	408,23
5843,1	0,78	328	297	289	36,29	408,23
6026,8	0,78	328	297	289	54,43	408,23
6200	0,78	339	297	289	54,43	408,23
6367,2	0,78	339	297	289	54,43	408,23
6521,6	0,78	339	297	289	54,43	408,23
6673,2	0,78	339	297	289	54,43	453,59
6833	0,78	339	297	289	54,43	453,59
6976,2	0,78	339	297	289	54,43	453,59
7091	0,98	339	297	289	54,43	453,59
7199,4	0,98	339	297	289	54,43	453,59
7300,2	0,98	339	297	289	54,43	453,59
7416,5	0,98	339	297	289	54,43	453,59
7539,9	0,98	339	297	289	54,43	408,23
7701	0,98	328	297	289	54,43	408,23
7812,6	0,98	328	297	289	54,43	408,23
7973,8	0,78	328	297	289	54,43	408,23
8111,3	0,78	328	297	289	54,43	408,23
8296,5	0,78	328	297	289	54,43	408,23
8411,7	0,78	339	297	289	54,43	408,23
8531,3	0,78	328	297	289	54,43	408,23
8668,8	0,78	339	297	289	54,43	408,23
8814	0,78	339	297	289	36,29	408,23



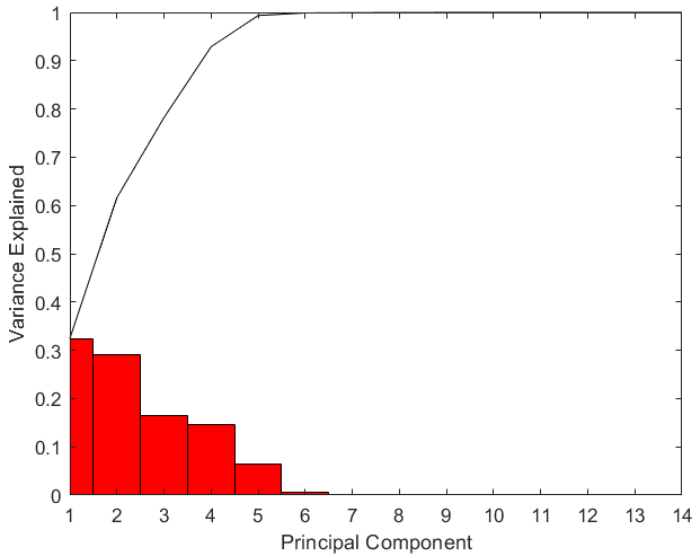


Figure 55: a) Complex CSTR process data generated using random seed 50 and a minimum steady state period of 100 hours b) Pareto chart of the complex CSTR data

Table 24: SSD tuning for the complex dataset

Window Size (*n*)	Variables	Threshold (θ_{ss})	Significance (α)
250	5 PCs	0.64	0.01%

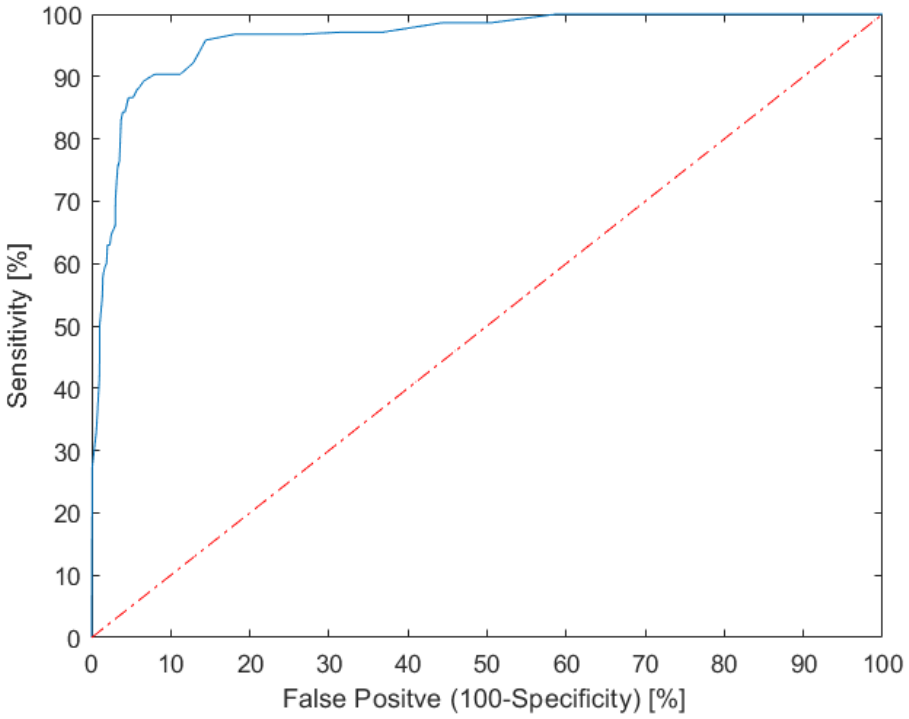


Figure 57: ROC curve obtained on complex CSTR dataset at tuning settings described by Table 24, achieving an AUC of 0.963

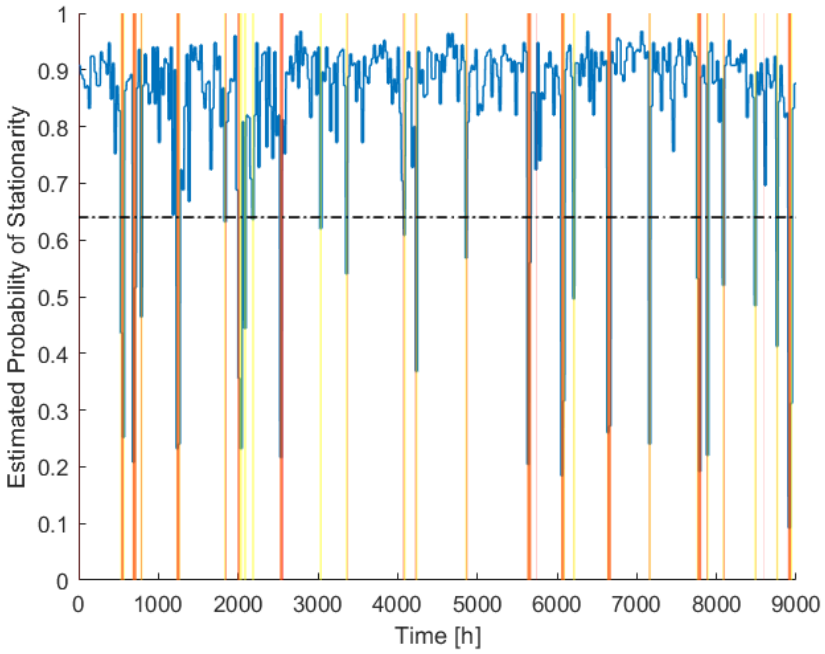


Figure 56: Visual SSD results obtained on complex CSTR simulation data

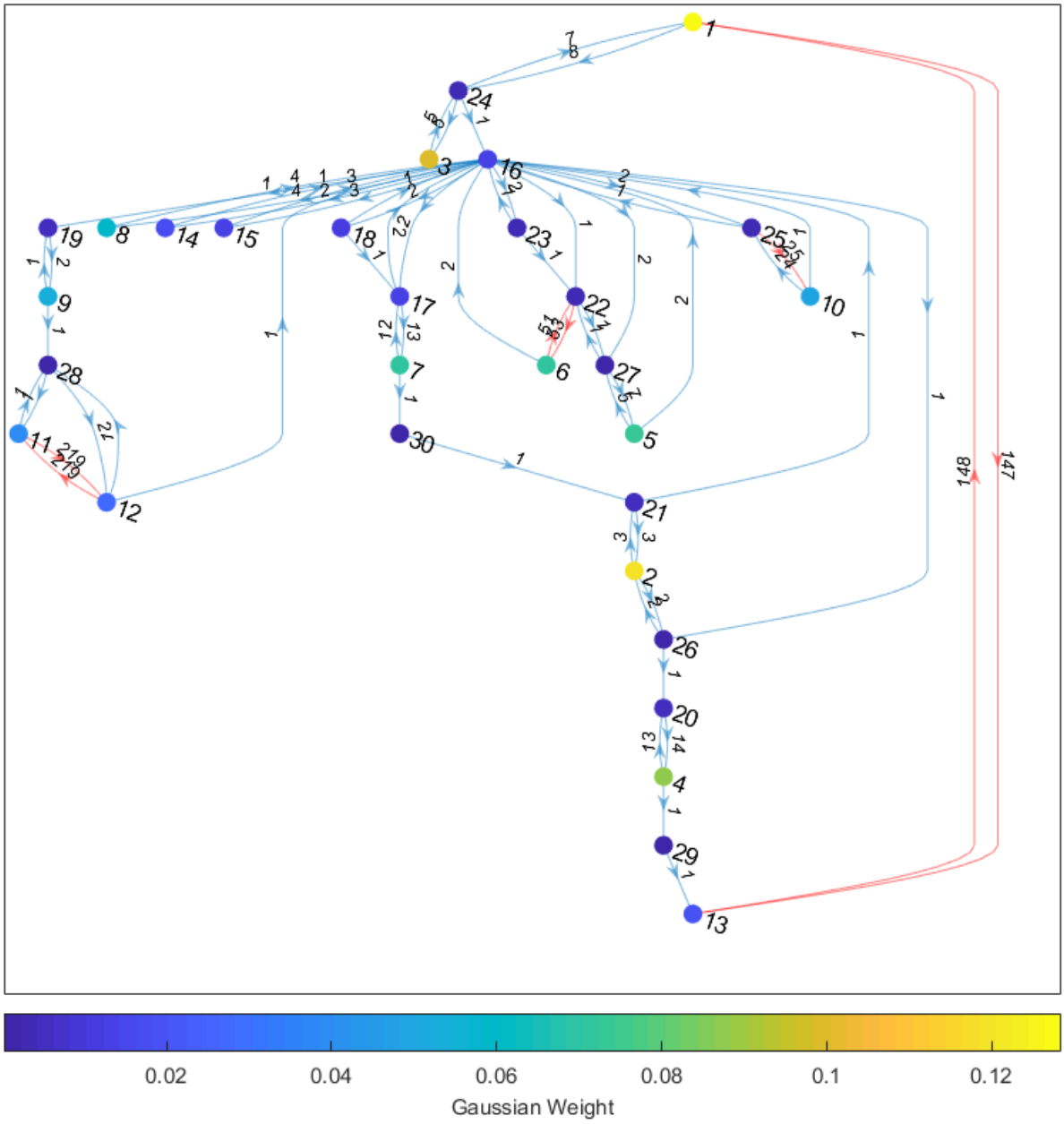


Figure 58: Graph of Figure 41 dataset, however with Euclidean distance between modes means not accounted for (rather MATLAB hierarchical layered method)

Table 25: Confusion Matrix obtained when analyzing the complex simulated data including transients

		Identified State Label															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0
True State Label	1	14325	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30
	2	0	10037	0	0	0	0	0	0	0	0	0	0	0	0	0	162
	3	0	0	8345	0	0	0	0	0	0	0	0	0	0	0	0	16
	4	0	0	0	8199	0	0	0	0	0	0	0	0	0	0	0	33
	5	0	0	0	0	6846	0	0	0	0	0	0	0	0	52	0	51
	6	0	0	0	0	0	7238	0	0	0	0	0	0	0	1	0	26
	7	0	0	0	0	0	0	6920	0	0	0	0	0	0	0	26	2
	8	0	0	0	0	0	0	0	4359	0	0	0	0	0	0	0	5
	9	0	0	0	0	0	0	0	0	4321	0	0	0	0	27	0	16
	10	0	0	0	0	0	0	0	0	0	3691	0	0	0	0	0	27
	11	0	0	0	0	0	0	0	0	0	0	4094	0	0	18	0	3
	12	0	0	0	0	0	0	0	0	0	0	0	2263	0	0	0	0
	13	0	0	0	0	0	0	0	0	0	0	0	0	1620	2	0	3
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	1151	0	0
	15	0	0	0	0	0	0	15	0	0	0	0	0	0	2	817	7
	0	46	114	69	69	47	12	85	45	35	18	106	244	58	3291	429	583

Table 26: Confusion Matrix obtained when analyzing the complex simulated after performing SSD

		Identified State Label															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0
True State Label	1	14321	0	0	0	0	0	0	0	0	0	0	0	0	0	0	34
	2	0	9958	0	0	0	0	0	0	0	0	0	0	0	0	0	241
	3	0	0	8342	0	0	0	0	0	0	0	0	0	0	0	0	19
	4	0	0	0	8171	0	0	0	0	0	0	0	0	0	0	0	61
	5	0	0	0	0	6851	0	0	0	0	0	0	0	0	0	0	98
	6	0	0	0	0	0	7236	0	0	0	0	0	0	0	0	0	29
	7	0	0	0	0	0	0	6926	0	0	0	0	0	0	0	18	4
	8	0	0	0	0	0	0	0	4355	0	0	0	0	0	0	0	9
	9	0	0	0	0	0	0	0	0	4322	0	0	0	0	0	0	42
	10	0	0	0	0	0	0	0	0	0	3682	0	0	0	0	0	36
	11	0	0	0	0	0	0	0	0	0	0	4089	0	0	0	0	26
	12	0	0	0	0	0	0	0	0	0	0	0	1985	0	0	0	278
	13	0	0	0	0	0	0	0	0	0	0	0	0	1573	0	0	52
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	1107	0	44
	15	0	0	0	0	0	0	17	0	0	0	0	0	0	0	820	4
	0	42	93	71	70	55	17	78	47	36	16	104	14	7	90	192	4319

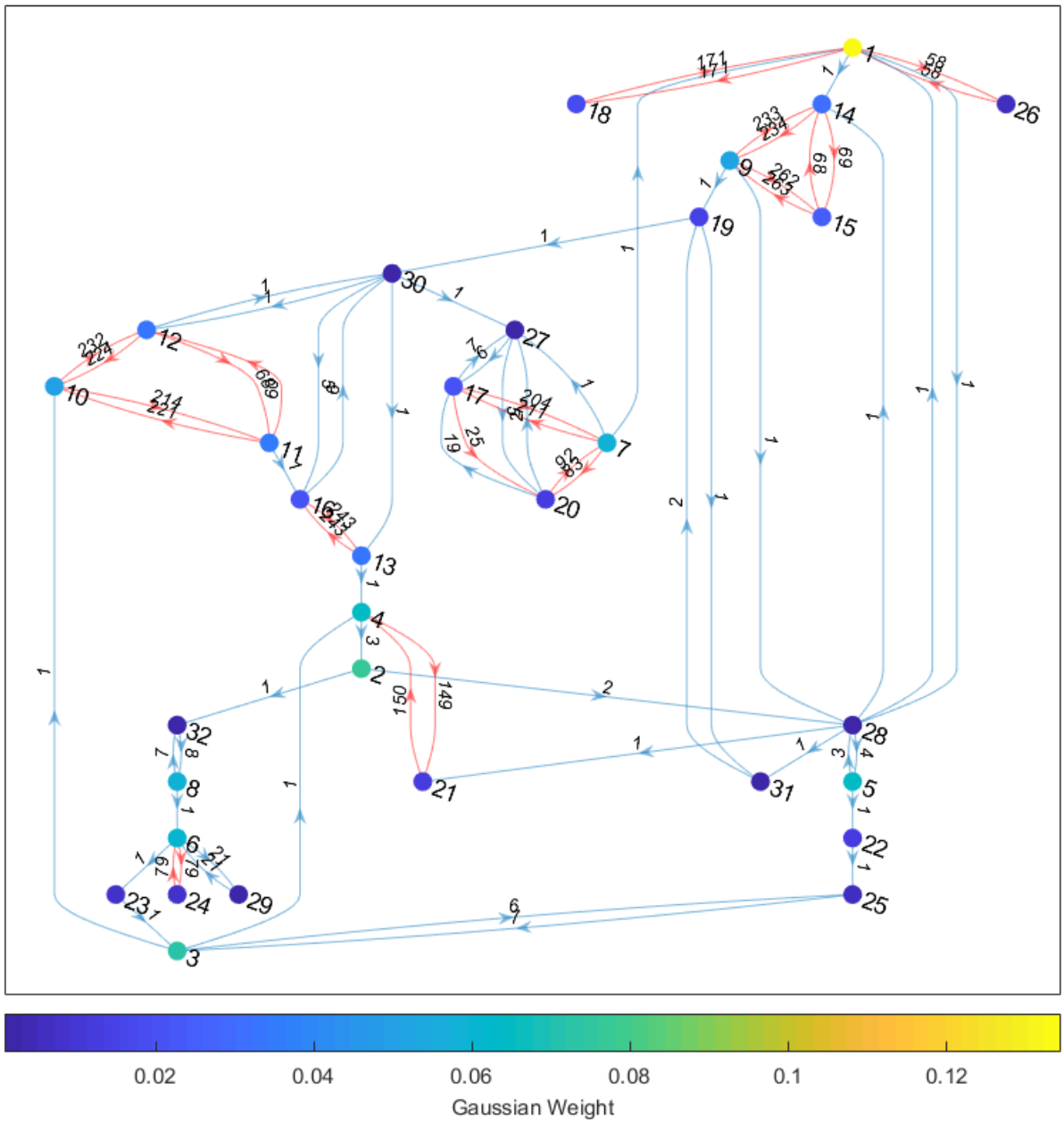


Figure 59: Graph of Figure 42 dataset, however with Euclidean distance between modes means not accounted for (rather MATLAB hierarchical layered method). Red edges indicate merged modes

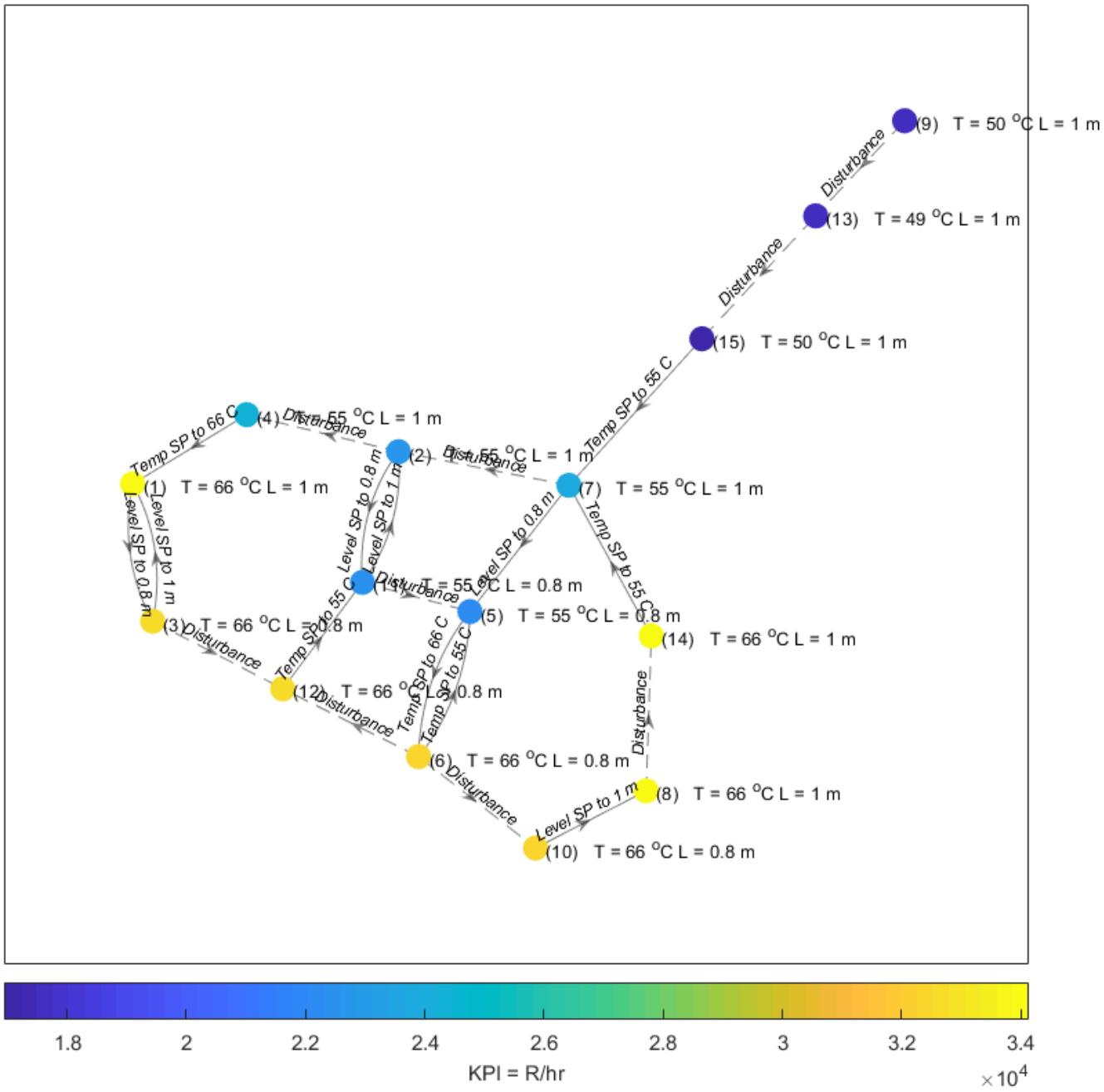


Figure 60: Ground truth state map for complex CSTR dataset

APPENDIX C: INDUSTRIAL DATA ANALYSIS

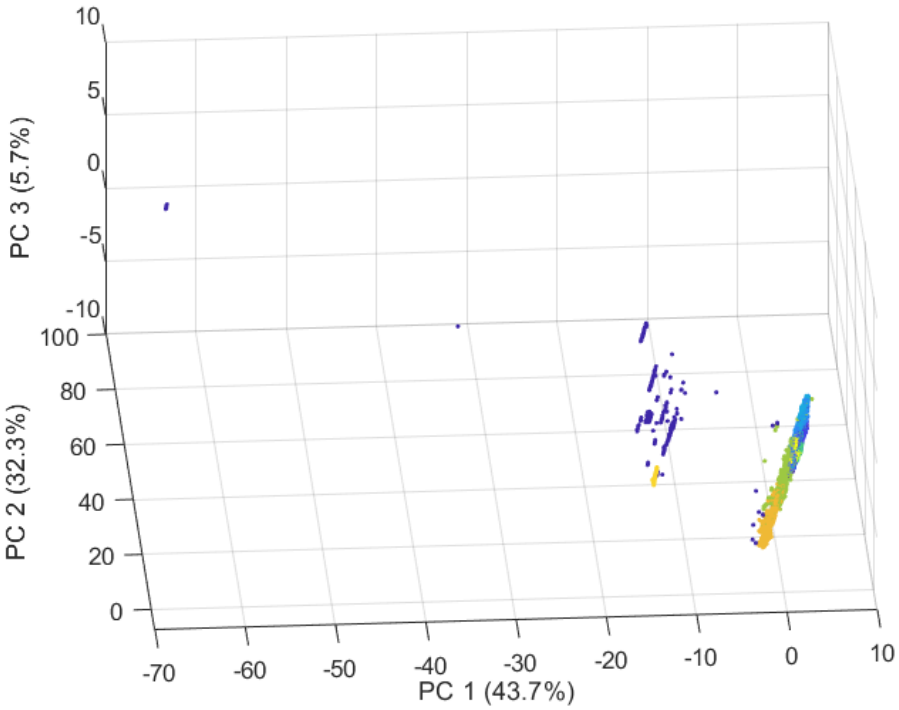


Figure 62: PC space of the milling circuit containing all data

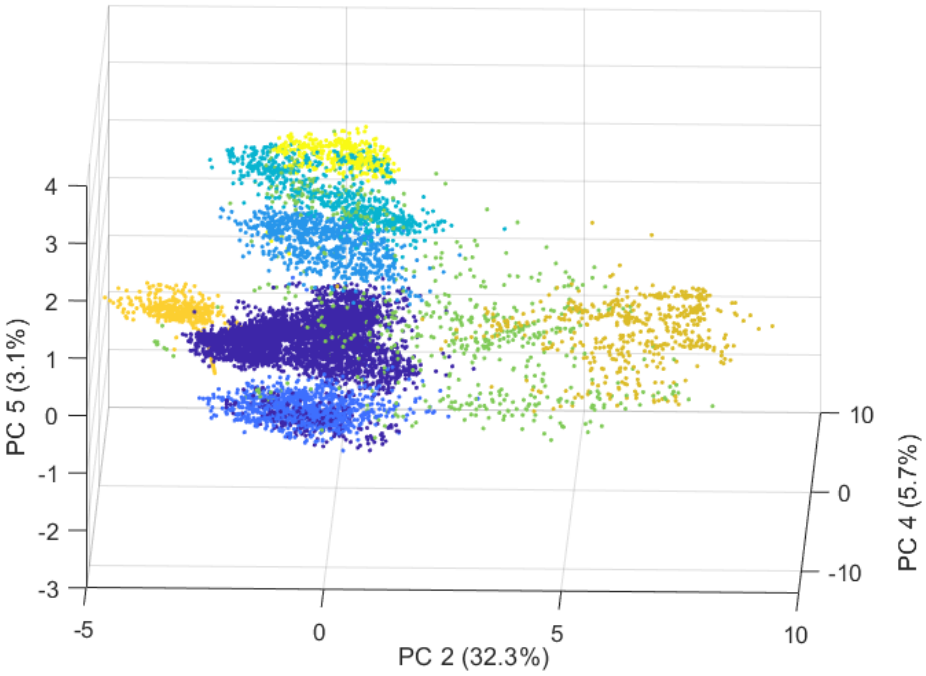


Figure 61: Adjusted milling circuit clustering

Table 27: Details of Milling Circuit variable identifying indices

Variable Index	Variable Description	Variable Index	Variable Description
1	Hydrocylone 1 Pessure	**34**	Mill Bearing Temperature 15
2	Sump Level	**35**	Mill Bearing Temperature 16
3	Sump Flow Promoter	**36**	Mill To Screen Flow (P-01)
4	Sump Concentration 1	**37**	Mill Lube Heat Exchanger Temperature 1
5	Sump Concentration 2	**38**	Mill Lube Heat Exchanger Temperature 2
6	Sump Flow Collecter	**39**	Mill Lube Heat Exchanger Temperature 3
7	Sump Total Mass Flow	**40**	Mill Lube Heat Exchanger Flow
8	Pump Current	**41**	Mill Lube Tank Temperature 2
9	Pump 1 Flow	**42**	Mill Lube TankTemperature 1
10	Pump 2 Total Mass Flow (P-02)	**43**	Mill Lube Pump Pressure 1
11	Mill Power To Load Ratio	**44**	Mill Lube Pump Pressure 2
12	Mill Load 1	**45**	Mill Lube Pump Flow 9
13	Mill Load 2	**46**	Mill Lube Pump Flow 10
14	Mill Motor Power	**47**	Mill Lube Pump Pressure 3
15	Mill Motor Temperature 1	**48**	Mill Lube Pump Pressure 4
16	Mill Motor Temperature 2	**49**	Mill Lube Pump Flow 1
17	Mill Motor Temperature 3	**50**	Mill Lube Pump Flow 2
18	Mill Motor Temperature 4	**51**	Mill Lube Pump Flow 3
19	Mill Motor Temperature 5	**52**	Mill Lube Pump Flow 4
20	Mill Bearing Temperature 1	**53**	Mill Lube Pump Flow 5
21	Mill Bearing Temperature 2	**54**	Mill Lube Pump Flow 6
22	Mill Bearing Temperature 3	**55**	Mill Lube Pump Flow 7
23	Mill Bearing Temperature 4	**56**	Mill Lube Pump Flow 8
24	Mill Bearing Temperature 5	**57**	Mill Lube Pump Pressure 5
25	Mill Bearing Temperature 6	**58**	Mill Lube Pump Pressure 6
26	Mill Bearing Temperature 7	**59**	Mill Lube Pump Pressure 7
27	Mill Bearing Temperature 8	**60**	Mill Lube Pump Pressure 8
28	Mill Bearing Temperature 9	**61**	Mill Lube Pump Pressure 9
29	Mill Bearing Temperature 10	**62**	Mill Lube Pump Pressure 10
30	Mill Bearing Temperature 11		
31	Mill Bearing Temperature 12		
32	Mill Bearing Temperature 13		
33	Mill Bearing Temperature 14		

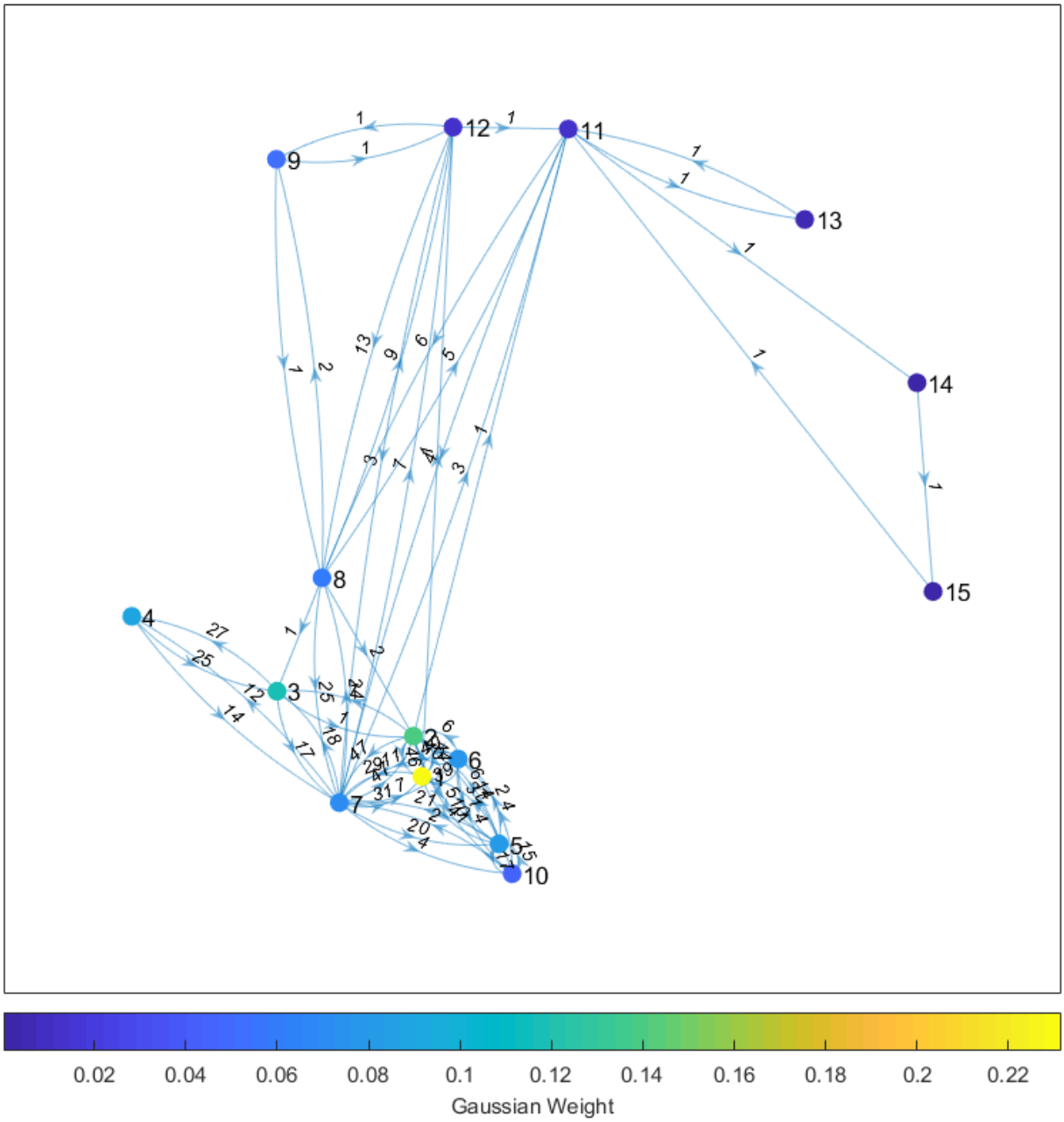


Figure 63: Graph assisting the GMM refining procedure on milling circuit data based on the data sequence, Euclidean distance between mode means, and Gaussian weights (node colour).

APPENDIX D: CSTR MODEL PARAMETER DESCRIPTION

Table 28: CSTR Simulation Parameters

Description	Value
CSTR Volume (V)	1.89 m^3
CSTR Radius	0.78 m
Heat Transfer Coefficient (UA)	7262 $\frac{kcal}{hK}$
Heat of Reaction (ΔH_{rx})	-20.013 $\frac{kcal}{kmol\ A}$
Specific Heat Capacity of propylene oxide (Cp_a)	35 $\frac{kcal}{kmol\ K}$
Cp_b	18 $\frac{kcal}{kmol\ K}$
Cp_C	46 $\frac{kcal}{kmol\ K}$
Cp_m	19.5 $\frac{kcal}{kmol\ K}$
Cooling Water Temperature (T_c)	289 K
Reaction Rate Constant (k_o)	$16.96 \times 10^{12}\ \frac{1}{h}$
Activation Energy	18012 $\frac{cal}{mol}$
Gas Constant (R)	1.987 $\frac{cal}{mol\ K}$
Molar density of propylene oxide (ρ_A)	14.8 $\frac{kmol}{m^3}$
ρ_B	55.3 $\frac{kmol}{m^3}$
ρ_M	24.7 $\frac{kmol}{m^3}$
Propylene Glycol Cost Parameter (p_{glyc})	R 22/kg
$p_{prop\ oxide}$	R 8.5/kg
$p_{cooling}$	R 195/m^3

Table 29: CSTR Start-up Parameters

Description	Value
C_a	0 $\frac{kmol}{m^3}$

C_{ao}	2.9 $\frac{kmol}{m^3}$
C_b	55.3 $\frac{kmol}{m^3}$
C_{bo}	36.3 $\frac{kmol}{m^3}$
C_c	0 $\frac{kmol}{m^3}$
C_m	0 $\frac{kmol}{m^3}$
C_{mo}	3.63 $\frac{kmol}{m^3}$
V_{in}/V_{out}	12.5 $\frac{m^3}{h}$
Reactor Temperature (T)	297 K
Reactor Feed Temperature (T_o)	297 K
Reactor Cooling Water Temperature (T_c)	289 K
Cooling Water Flowrate (m_c)	453.6 $\frac{kmol}{h}$

It should be noted that each simulation contains the same initial start-up mode or in other words achieves the same initial steady state for a short duration.

Table 30: CSTR Controller Parameters

Description	Simple Simulation	Complex Simulation
Temperature Controller Integral Time	2	0.5
Temperature Controller Proportional Gain	-10	-10
Level Controller Integral Time	0.2	0.2
Level Controller Proportional Gain	-20	-20

Table 31: Process Noise and Measurement Noise Parameters

Description	Process Noise (σ_e^2)	Autoregressive Coefficient (ϕ)	Measurement Noise (σ_m^2)
Inlet Flowrate (V_{in})	--	--	0.5
Inlet Concentration (C_{ao})	--	--	2.5×10^{-6}
C_{bo}	--	--	2.5×10^{-6}
C_{mo}	--		2.5×10^{-6}

Level	--	--	2×10^{-4}
Outlet Flowrate (V_{out})	--	--	0.5
C_a	--	--	2.5×10^{-6}
C_b	--	--	2.5×10^{-6}
C_c	--	--	2.5×10^{-6}
C_m	--	--	2.5×10^{-6}
Feed Inlet Temperature (T_o)	0.05	0.7	0.5
CSTR Temperature (T)	--	--	5×10^{-3}
Cooling Water Temperature (T_c)	0.05	0.7	0.5
Cooling Water Flowrate (m_c)	--	--	0.5
F_{ao}	0.005	0.7	--
F_{bo}	0.005	0.7	--
F_{mo}	0.005	0.7	--

Table 32: First Order Disturbance Response Parameters

Description	Time Constant
Methanol Molar Inlet Flowrate (τ_M)	10
Water Molar Inlet Flowrate (τ_B)	10
Cooling Water Temperature (τ_{T_c})	8
Feed Inlet Temperature (τ_{T_c})	8
Level SP Filter Time Constant	4

Table 33: Low and High Settings of Certain Input Variables for Random Multimodal Data Generation

Description	Low	High
Methanol Molar Inlet Flowrate	36.3 $\frac{kmol}{h}$	45.4 $\frac{kmol}{h}$
Water Molar Inlet Flowrate	408.2 $\frac{kmol}{h}$	453.6 $\frac{kmol}{h}$

Cooling Water Temperature	289 K	289 K
Feed Inlet Temperature	297 K	297 K
Level SP	0.78 m	0.98 m
Temperature SP	328 K	339 K

Table 34: CSTR Random Multimodal Simulation Details

Simulation Details	Randomising Seed	Minimum Steady State Time	Simulation Duration
Simple CSTR Simulation (6 modes)	60	200 h	8000 h
Complex CSTR Simulation (15 modes)	50	100 h	9000 h

CSTR Assumptions:

- No mass transfer effects to the atmosphere
- Perfect mixing
- Constant physical properties
- Negligible shaft work
- Single irreversible reaction